NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

MIGRATING FROM MAINFRAMES TO CLIENT-SERVER SYSTEMS

by

Todd L. Barnum

September 1995

Thesis Advisors: James C. Emery
Myung W. Suh

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE			Form Approved OMB No. 0704
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	2. REPORT DATE September 1995	3. REPORT TYPE AND DATES COVERED Master's Thesis, Final	
4. TITLE AND SUBTITLE MIGRATING FROM MAINFRAMES TO CLIENT-SERVER SYSTEMS			5. FUNDING NUMBERS
6. AUTHOR(S) Todd L. Barnum, USN			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSORING/MONITORING AGENCY REPORT NUMBER
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government.			
12a. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited			12b. DISTRIBUTION CODE
13. ABSTRACT *(maximum 200 words)* The prevailing trend within the computer industry is to downsize information systems. This quite often entails migrating an application from a centralized mainframe environment to a distributed client-server system. Navy IS managers are often given the mandate to downsize all information systems with out much consideration for the framing issues of strategic planning and Business Process Reengineering(BPR). The decision to migrate off a mainframe is a difficult one to assess, as the issues come from a broad spectrum of sources. This thesis analyzes the mangment issues, associated with this migration, and looks at both the role of BPR and some of the options to migrate applicatons off the mainframe to client-server systems. This thesis also aims at educating the Navy IS manager regarding the new client-server computing model as well as providing background to the management practice of BPR.			
14. SUBJECT TERMS Business Process Reengineering, Downsizing Information Systems, and Client-server Systems.			15. NUMBER OF PAGES 90
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std. 239-18

Approved for public release; distribution is unlimited.

MIGRATING FROM MAINFRAMES TO CLIENT-SERVER SYSTEMS

Todd L. Barnum
Lieutenant, United States Navy
B.A., University of Hawaii, 1984

Submitted in partial fulfillment
of the requirements for the degree of

MASTER OF SCIENCE IN INFORMATION TECHNOLOGY MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 1995

Author: ______________________
Todd L. Barnum

Approved by: ______________________
James C. Emery, Thesis Advisor

____
Myung W. Suh, Associate Thesis Advisor

____
Reuben Harris, Chairman
Department of Systems Management

ABSTRACT

The prevailing trend within the computer industry is to downsize information systems. This quite often entails migrating an application from a centralized mainframe environment to a distributed client-server system. Navy IS managers are often given the mandate to downsize all information systems without much consideration for the framing issues of strategic planning and Business Process Reengineering (BPR). The decision to migrate off a mainframe is a difficult one to assess, requiring the consideration of a broad spectrum of issues. This thesis analyzes the management issues associated with this migration, and looks at both the role of BPR and some of the options to migrate applications off the mainframe to client-server systems. This thesis also aims at educating the Navy IS manager regarding the new client-server computing model as well as providing background to the management practice of BPR.

TABLE OF CONTENTS

I. INTRODUCTION

A. PURPOSE

The purpose of this thesis is to provide a management overview of the issues associated with migrating from a mainframe computer environment to a client-server system. The decision to make this transition is multifaceted, and, given today's complex computing environment, a very difficult one to gauge. Navy Information Systems(IS) managers are faced with the hardship of having to do more with less, while confronting an uncertain and ever-changing future. Consequently, knowing up front the implications and risks associated with moving from a mainframe computer to a client-server system will enable Navy IS managers to better judge whether this transition is a worthy pursuit.

B. THE NEED FOR CLIENT-SERVER COMPUTING

We live in a period marked by rapid change. Change is such a hot topic that most universities have developed curricula's dealing with "Change Management" and its implications on the business environment. Consultants specializing in change management make large sums of money and are believed to be essential to any change initiative. Probably the sector of our society that has experienced the largest rate of change has been the computer industry. Computers are more pervasive today than they ever have been, and it would be hard to find an industry that could boast of being Information System (IS) independent. It was only fifteen years ago that desktop computers were introduced at which time they were associated with researchers and engineering types who spoke a language only they understood. This has all changed.

1. Global Economy

In the early and mid seventies the US. for the first time began to experience increased competition from foreign competitors. The Japanese and Germans were producing quality products and gaining sizable footholds in substantial sectors of the US

market. This increased competition from abroad caused the US to look inward and analyze its corporate structure and management practices.

Many US corporations were challenged to assess the way in which they conducted business. These assessments were driven by declining profits and increasing competition from foreign products. It did not take long for US corporations to realize that their corporate structures, which had served them so well throughout the industrial revolution and up through the early Eighties, were an impediment to their competitive well being. At a meeting of the Japan Society of New York, then Deputy Secretary of the Treasury Richard Darman said

> that bloated, risk averse, inefficient, and unimaginative large corporations make up an American business "corpocracy," and that this corporate bureaucracy was a key reason behind the decline of the United States' global competitiveness...[Ref. 1, p. 1]

The American concept of how industrialized work should be structured was dying. Senior executives in the US noticed that the Japanese and Germans had entirely different management styles, not to mention corporate structures. These realizations led to the initial rethinking of American corporate structure and how traditional work tasks were organized. Management began to focus on improving the design of work processes with an eye toward increased efficiency and customer satisfaction.

These shifts in management focus were the direct result of increased competition from abroad. No doubt the US would have continued business as usual without the external pressures applied on businesses. According to Peter F. Drucker, "[W]e had entered a period of change: the shift from the command-and-control organization, the organization of departments and divisions, to the information-based organization, the organization of knowledge specialists. [Ref. 2, p. 11]

2. Decentralization of the Work Center

Consequently corporations found that to remain competitive they needed to reorganize their corporate chart, which for many years had served as the backbone of

corporate life. Corporations had traditionally been organized in highly cohesive workcenters centered around departments. Management was centralized, and conducted itself in a top-down fashion. Those at the top yielded the power, made the decisions, and guided the corporate direction. Middle management had swollen to the point of gluttony, and the layers of supervision were suffocating. Consequently, the forces were in motion to create a more "horizontal organization" where management was exercised across and in teams rather than up and down. John Byrne in his article titled *The Horizontal Corporation* addresses these shifts by stating that:

> To change the fundamental way that work gets done in a corporation will take a different organizational model, the horizontal corporation. In the quest for greater efficiency and productivity American Corporations are beginning to redraw the hierarchical organization charts.... [Ref. 3, p. 76]

3. Corporate Reengineering

The next step in the evolution of the American corporation led to the management movement of reengineering. Reengineering was championed in the late Eighties and early Nineties by Michael Hammer and James Champy who authored *Re-engineering the Corporation.* Although they articulated the principles of reengineering and canonized its disciplines, the actual reengineering process had been ongoing across America for several years. Companies such as Hallmark, Taco Bell, and Bell Atlantic had gone through reengineering-like processes nearly a decade earlier. The results of these reengineering efforts attracted management theorists like Hammer and Champy who began to study the principles of these drastic change programs.

These two men were able to extrapolate from the experiences of these early pioneers some principles around which early definitions of the reengineering process were defined. They married up the pressing demands facing American businesses with the plausible solutions contained in the reengineering process. They argued that without an entire corporate overhaul American businesses were doomed to failure. Their

assertions were that American corporate structures were hostile to successful business functions and that to regain health:

> American managers must throw out their old notions about how businesses should be organized and run. They must abandon the organizational and operational principles and create entirely new ones. [Ref. 4, p. 1] Business reengineering meant starting all over, starting from scratch. [Ref. 4, p. 2]

Incorporated in this view Hammer and Champy saw the strategic role Information Technology (IT) could play in redesigning business processes. They went so far as to say that IT was quintessential to the survival and continued profitability of American Corporations. Information Technology held the capability to empower workers in the "new age" corporation that was team-oriented, customer focused, and delivered high quality products and services. Through the use of information technology corporations could increase coordination among various business components, and thus be more productive. Therefore American corporations viewed IT as the leveraging mechanism that would return the US to its role as the leading economic nation.

4. Role of Technology

Prior to the reengineering movement the computer industry was undergoing a revolution of titanic proportions. This revolution had its origins back to the early Eighties with the introduction of desktop computing. Since that inception, the computer industry - more specifically, the processing power being provided to end users - grew exponentially. No longer were computer capabilities reserved for those who worked in the "computer room." End users possessed processing power that enabled them to be more productive and less dependent upon IS department personnel for business solutions. This increased productivity and flexibility held tremendous potential for American corporations.

Nevertheless, desktop computing was not without its drawbacks. Early systems were cryptic in nature and relegated to those who could understand their unique languages. Moreover acquisitions costs were high, prohibiting widespread personal use.

For these reasons organizations were unable to capitalize on any gains in productivity they might have from desktop computing.

However these drawbacks began to dissipate as prices fell and ease of use increased. Processing capabilities of desktop machines increased dramatically while the accompanying acquisitions costs fell. Table 1 lists some of the improvements in desktop computing since the IBM desktop computer was introduced in 1981. These improvements cannot be understated, and have been a major contributor to the America's continued economic power.

Feature	1981	1991
CPU	8088	80486
Clock Frequency	4.77 MHz	50 MHz
MIPS	0.3	40.4
Number of Transistors	29,000	1,200,000
Floppy Disk Size	5.25 inches	3.5 inches
Floppy Disk Capacity	360 KB	1440 KB
Internal Disk Space	10 MB	640 MB

Table 1: IBM PC Improvements, After [Ref. 5, p. 35]

5. Heterogeneous Systems

Although the desktop revolution has been a blessing for end users, it has created some large problems for the IS world. Unlike the early Eighties, when there were very few computer platforms, there is now an inexhaustible number of options to choose from, and only those who stay abreast of the computer market can decipher the terms and associated functionalities. Fortunately this increased burden has more than been offset by lower priced systems.

The days of one dominant "anything" are over as consumers have a wide choice of computer products and capabilities. This wide selection of options has given rise to a

heterogeneous atmosphere that presents additional challenges to users. Ralph Sprague and Barbara McNurlin assert in their book *Information Systems Management in Practice* that:

> the goal today is not a single coherent network but rather finding a means to interface many dissimilar networks. [Ref. 6, p. 184] More and more organizations are seeing the need to tie together their islands of automation, seeking what the worldwide telephone system already provides: the ability for any telephone user to be connected with any other user. Connectivity means allowing users to communicate up, down, across, and out of an organization. [Ref. 6, p. 185]

6. Vendor Hype

The last major contribution to this chaotic atmosphere of change has been the role and influence of computer vendor hype. It is difficult to read any computer journal and not find hundreds of vendors touting their products as the solutions to all corporate computing needs. Vendor hype has led IS managers to falsely believe that the latest fad will solve all corporate computing problems. Likewise IS managers have prematurely subscribed to this belief regarding client-server technology and many have suffered dearly for it.

Like the reengineering movement, the statistics on the success of client-server implementations cover a very broad distribution. One computer article will claim that client-server implementations enjoy a success rate of around 60 percent, while another will claim a success rate of only 15 to 20 percent. This wide range of results makes it imperative that the Navy IS manager understand the implications of migrating to client-server systems and what factors increase the probability of success. Interestingly enough, most vendors were initially pushing client-server as a cost-savings mechanism but have since backed off from these claims.

The computing model is undergoing a major shift from a centralized mainframe environment to a decentralized client-server model. The issues mentioned above - the global economy, the decentralized workplace, the reengineering process, the falling

hardware costs, the heterogeneous computer systems, and the vendor hype - have been significant contributors giving rise to this new computing model.

C. THESIS OUTLINE

This thesis will provide an analysis of the issues associated with downsizing information systems. Clearly there is a transition underway within organizational IS shops, influencing IS managers to exchange the traditional mainframe for the new client-server architecture. However, the undercurrents causing this shift are often nebulous and hard to identify. Sifting through these mixed signals to arrive at a decision whether to downsize a system or application will be the central discussion of this thesis.

The content of this thesis will result from a literary review of current industry trends affecting the "downsizing" of information systems. The decision to migrate from mainframes to client-server, also called "downsizing," is double sided -- both management and technical in nature. This research will address both facets; however, the primary focus will be on the management issues while giving brief mention to some the technical aspects.

The goal of this thesis is to educate Navy IS managers regarding the complexity of the downsizing process and to provide the basic framework for a successful downsizing project. The topic of downsizing information systems is incredibly broad and any one section or chapter could be expanded into an entire thesis. The intention, though, is to present the major issues that a Navy IS Manager should be aware of when confronted with this decision.

After this introductory chapter, the second chapter will discuss the reengineering trends that have shaped American organizations over the recent past. The topic of reengineering is an important issue, as it has illicited broad consensus that organizations will become more profitable if key business processes are reengineered. From a management point of view this means reducing bloated bureaucracies to make way for smaller, more dynamic organizations and systems. Centralized IS departments with their closely guarded and largely inaccessible resources, were among the first entities to feel

the impacts of the reengineering wave. Recent reengineering projects have lauded the strategic role that information technology can play, and it is incumbent upon Navy IS managers to understand this critical role of IT.

The third chapter will focus on the downsizing effort itself. In light of Chapter Two's discussion of reengineering, the thrust will be analyzing how and under what situations the mainframe, and mainframe applications, should be downsized. One key point to be made is that there are clearly some situations in which the mainframe should not be downsized. This goes against popular media rhetoric where such phrases as "Shoot the mainframe" and "Don't Automate: Obliterate" are popular. In contrast, there are those who oppose this "shoot the mainframe" mentality and see a useful role for the mainframe in the modern IS organization. Discussion in this chapter will cover the architecture, role, advantages and disadvantages of operating mainframe computers, and a look at mainframe applications that are ripe for downsizing to smaller systems and some of the risks involved.

The fourth chapter will be devoted to the client-server architecture. Since there is a prevailing tendency throughout the IS community to subscribe to this latest technology it is essential that the Navy IS manager understand the major management issues associated with deploying this architecture. Moreover, there are some critical issues that must be addressed when migrating to client-server systems. These issues are frequently discussed in articles warning IS managers of the hidden pitfalls associated with migrating to a client-server system, but are not often mentioned by vendors pushing the client-server band-wagon. These issues revolve primarily around the hidden costs of support, training, and network management.

Client-server computing offers many advantages to organizations wishing to reengineer their business processes. Of equal importance is understanding the environment and organizational demands that have given rise to the need for distributed computing. Knowing these forces will help the Navy IS manager better discern when and under what conditions the transition to client-server is a wise choice.

The fifth chapter will be a summary of the thesis and the author's beliefs regarding the management imperatives associated with the downsizing movement. Navy IS managers who have an appreciation for the issues and obstacles in downsizing information systems will have a head start on avoiding its many pitfalls. Client-server computing is not a passing computer fad, but is the new computing model. Therefore it is essential that Navy IS managers understand it to the fullest extent possible.

II. BUSINESS PROCESS REENGINEERING

A. AMERICAN CORPORATE HISTORY

America has over 200 years of business history. Most companies can trace their work style and organization structure back to the prototype factory described by Adam Smith in *The Wealth of Nations*. As a philosopher and economist he realized that breaking work down into its simplest tasks would produce large increases in worker productivity. Smith's observations led to specialized workers performing a single task in what is today known as the assembly line. Workers were trained to perform tightly controlled procedures, and management was installed to monitor and measure worker performance. As a result of specialized labor, worker productivity increased tenfold. Over time American companies perfected this principle of work specialization to a science.

1. The Organizational Chart

One of the byproducts that of the specialization of labor was the structuring of personnel by departments, with each department having responsibility for one piece of the production process. Out of this structure organizations eventually grew into "stovepipe" bureaucracies characterized by layers of management put in place to monitor the performance of workers. This top-down management style became the model for industrialized nations and rarely did an organization deviate from this framework.

Modern organizational charts reflect this highly centralized top-down framework. For most companies, this framework provided a foundation around which business strategies were built and workers managed their careers. Workers were organized around departments and looked inward toward their department or upward toward their boss, but few rarely looked beyond the department boundaries.

2. Loss of Corporate Vision

Eventually organizations lost sight of their true mission to provide quality products and services. The focus shifted to the work being done, and management concentrated its efforts on refining and maximizing production. Rarely did management question whether its efforts or processes were meeting the needs of customers. All was okay as long as the profit sheet said so. Gradually, and unbeknownst to management, these bureaucratic structures had become an obstacle to corporate goals. Eventually good products and quality service were lost in management's efforts to increase productivity, since it was shown that higher productivity equated to greater profits. Two members from the Gartner Group research firm acutely stated as recently as 1994 that:

> despite a decade or more of restructuring, downsizing and applying new information technology, many US. companies remain uncompetitive and unable to cope with growing economic globalization. Many executives are realizing that their organizational structures, job descriptions and product work flows were implemented in response to the business priorities of a different era. [Ref. 7, p. 1]

This different era was the Industrial Age. In this era it was common for the vice president of a department to have worked himself to the top, learning every aspect of the department. Eventually this specialization of labor became a huge liability as personnel within departments put departmental interests before corporate interests. Sub-cultures were established around department and division lines. Product development and refinement was removed far away from the customer and placed in isolated research and development centers that never saw or heard from customers. During the later part of the Industrial Age these organization structures were cemented into place through automation. [Ref. 8, p. 12] As a result, most companies suffered from the following four characteristics:

- An organizational chart whose functional boundaries represent territories rather than lines of communication.
- A reward system that measures only individual effort and gives no visibility to cross-functional collaboration.

- Autonomous business units that complain that the source of their inefficiencies is the ineptitude of other departments.
- A perception from customers of a lack of responsiveness to their needs. [Ref. 9, p. 1]

All of these symptoms were manifestations of management that had lost focus. Attempts to resuscitate management to improve productivity and competitiveness had little affect at rescuing American corporations. These business structures that served the industrial era of mass production were dying. It wasn't until the early Eighties that these struggles facing corporate America became more evident to management consultants.

B. REENGINEERING SURFACES

Two consultants who noticed the ineffectiveness of the present structure and the need for change were Michael Hammer and James Champy. They discovered this ineffectiveness by noting a few companies that had drastically improved their performance, productivity, and profits. These improvements were not the result of new products or markets but instead resulted from major alterations of their business processes. These companies had not only survived global competition but had even expanded their customer base to include foreign markets.

Hammer and Champy later authored *Reengineering the Corporation* published in 1993 that made a big impact among management circles. In this work they provided insightful background into the forces of change that led to American companies reengineering their business processes. From their studies of reengineered corporations Hammer and Champy credit three forces that brought about the need for industry-wide business process reengineering (BPR).

1. Fundamental Change in Buyer-Seller Relationship

First, there was a fundamental shift in the buyer-seller relationship. Customers were now in charge, and sellers could no longer ignore customer's demands. Previously, customers bought what was offered because they didn't have much selection to choose from. As the market offered more products customers became more knowledgeable and

began to discriminate among products. With increased knowledge these customers could no longer be cast in the same mold with all other customers. They shopped for products that were tailored to their specific needs and conformed to their delivery and payment schedules.

Today customers know what products they want, at what price, and under what terms. In almost all respects it is the customer who determines the deal. Consequently, customers don't take the time or wish to deal with businesses that are not responsive to their needs. [Ref. 4, p. 18]

2. Increased Competition

The second force that brought about the need for industry-wide change was the increase in competition throughout the United States. With the disappearance of trade barriers, national trading turf was no longer off limits to foreign competitors. For example, American automobile manufacturers had to compete with the likes of Honda, Toyota, and Mercedes. "Adequate" was no longer good enough in the face of keen competition and increased customer demands. [Ref. 4, p. 21]

A good illustration of corporate America's unresponsiveness to customer shifts occurred in the early Seventies during the oil crisis. US. drivers demanded cars that achieved higher mileage rates than those offered by most American manufacturers. American auto-makers were unresponsive to these customer demands and chose to continue to manufacture autos that traveled 12 to 16 miles per gallon. Japanese auto-makers, who had previously been derided for their smaller autos, were in a position to capitalize on this unmet demand. American drivers who made the switch to Japanese autos for economic reasons continued to buy Japanese autos in unprecedented numbers.

3. Change Management

Finally, Hammer and Champy noticed that "change" was now a constant. The corporate world could not continue to exist in a static state of maximum productivity. Customer demands changed rapidly and to keep pace, companies needed flexibility to

adapt to these sudden shifts. Companies that were static or unresponsive found themselves losing their customer base. Furthermore, product life cycles changed. Car retention went from an average of seven years down to three years. [Ref. 4, p. 23] The personal computer of today, unlike its predecessor, will be technologically obsolescent in less than a few years. Businesses have given ear to customer needs, and have encouraged feedback on products and services. Researchers from the Gartner Group have summarized these transitions by stating that:

> Today, the corporate world is finally trying to respond to the demands of customers who have changed their expectations and definitions of service. Corporate restructuring, business process reengineering and flattening of the organization are all attempts to dismantle the bureaucracy which had been diverting the focus of companies from their customer issues. The old functional divisions among departments must give way to a new organizational chart based on channels of communication that cross functional lines. [Ref. 9, p. 1]

C. BUSINESS PROCESS REENGINEERING (BPR)

It was from these market shifts that business process reengineering (BPR) was born. Vendors were faced with customers demanding better products that were more responsive to their needs and at lower prices. Meanwhile companies were struggling to survive, and were pushed to the point of seeking drastic measures to regain a place in the market.

1. Reengineering Defined

Hammer and Champy defined Business Process Reengineering (BPR) as "the fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in critical contemporary measures of performance, such as cost, quality, service, and speed." [Ref. 4, p. 32] This definition is amplified by researchers from the Gartner Group who stated that:

> Business process reengineering is the analysis and radical redesign of an orgaiazation--not just business processes, but management systems, job definitions, organizational structures, beliefs and behaviors as well-- in

> an attempt to improve performance to meet contemporary requirements. BPR is not new; it comprises many traditional disciplines from industrial engineering, systems analysis and information engineering. [Ref. 7, p. 1]

These two definitions indicate that BPR efforts focus on the rethinking of work processes. To do so an organization must erase its mental model as to the assumed way in which work is accomplished. BPR assumes that those involved will aim for radical redesign and not incremental changes. Creative thinking is often the vehicle that allows for these types of changes to be accomplished.

In many respects, BPR is an attempt to reassemble the work processes that Adam Smith had previously disassembled. Key candidates for reengineering are those interdepartmental processes that can be consolidated into one process, thereby minimizing both the number of hand-offs and processing time. Mike Hammer states that:

> work processes should be organized around outcomes and not tasks. This principle says to have one person perform all the steps in a process. Design that person's job around an objective or outcome instead of a single task. [Ref. 10, p. 108]

BPR is ambitious, analytical, and creative. If performed correctly it will allow an organization to overhaul and simplify its job processes and organizational structure. The promises of BPR are great and successful reengineering efforts will most likely ensure continued market competitiveness.

2. What BPR Is Not

It may be easier to understand BPR by looking at what it is not. BPR is fundamentally different from the "total quality" initiatives that have been offered as the answer to what ails corporate management. Unlike the quality initiatives, BPR seeks ambitious results through drastic measures and creative thinking. Quality initiatives work within the established framework attempting to improve one aspect of an organization such as employee morale, work environment, or management-employee relationships. BPR, on the other hand, goes beyond organizational modifications and attempts to force an organization to stretch itself by thinking beyond the corporate structure.

Quality initiatives failed to accurately identify the causes of corporate America's decline and were therefore poor solutions. These initiatives held the view that performance problems were tied to "inadequate" workers, who were poorly trained, poorly motivated, and were encumbered with too many responsibilities. Management's response was more training, more bonuses, and fewer responsibilities. [Ref. 11, p. 3]

Unfortunately, quality initiatives addressed problems in departments as stemming from the organization's structure. This led to the thinking that corporations were not structured right and coordination problems were the result of unskilled employees. Hammer and Champy concluded that these assessments, levied by the quality movement, were wrong and that the real problems stemmed from the work itself and how the processes were engineered. Table 2 lists some of the major differences between the "quality initiatives" and BPR.

	Qualtiy Initiative	**BPR**
Degree of change	Incremental	Radical
Starting point	Existing process	Clean slate
Scope/focus	Narrow	Broad
Risk	Moderate/Low	High
Goals	Small, many	Outrageous
Role of IT	Incidental	Key

Table 2: The Differences Between BPR and Quality Initiatives, After [Ref. 7, p. 3]

As computing became more widespread many equated BPR with automation. Companies that merely automated existing processes did so thinking they would improve corporate performance. In most cases automation merely "paved the cow paths," and added very little value. Automating a bad process only tended to make matters worse, since the processes that needed to be reengineered were cemented into existence.

3. Goals of BPR

Basically BPR aims at task integration or what can also be referred to as process compression. Task integration involves identifying those processes that are performed by multiple people exchanging numerous hand-offs and visualizing a process where only one or very few people accomplish the entire task. The best candidates for reengineering are processes that are handed off across interdepartmental lines. Reengineering these "assembly-line" tasks will involve removing the many queues that the work passes through on its way to completion.

The quality initiatives were not bad programs, but alone they were incapable of producing the required change necessary for companies to reorganize into a competitive posture. Many believe that when implemented in conjunction with BPR the quality initiatives offered the sought-after results. Understanding when to apply BPR techniques and/or quality initiatives is the determination that must be gauged in light of the desired change.

Traditionally only managers were allowed to make decisions and employees were hired to do the work. However, once a number of tasks are integrated into one process the employee will need to have the power to control the decision-making surrounding the process. Without the authority over the newly integrated task the employee will be hindered from task completion until a manager is available, informed of the factors involved, and makes a decision.

Empowering the employee will produce positive consequences. The employee will have a greater sense of ownership over the work being performed since he or she will be responsible for the entire process from cradle to grave. In addition, this process will be accomplished in less time than the original series of processes with its many queues. [Ref. 12, p. 12] Figure 1 contrasts a process before and after it has been reengineered.

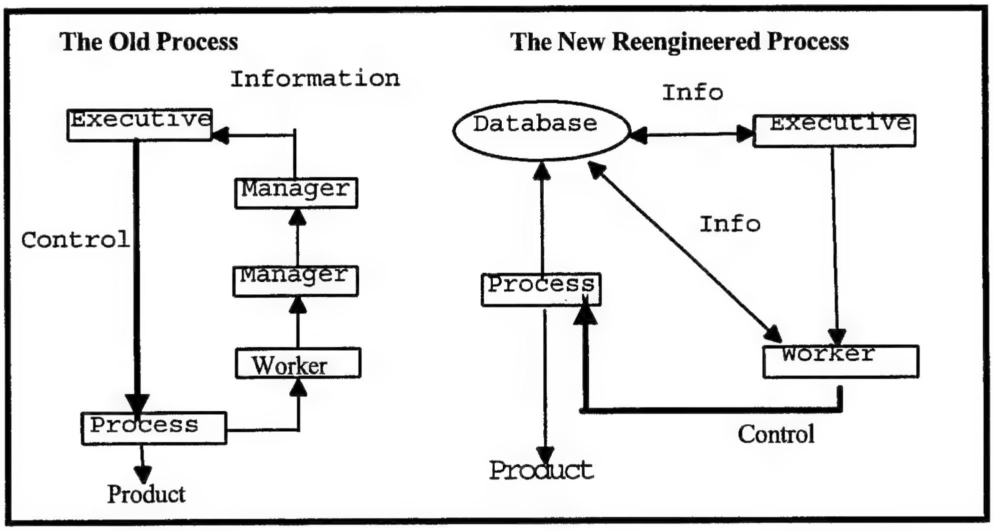

Figure 1: Control Systems: Old and New, After [Ref. 13, p. 24]

D. SUCCESSFUL BPR

Although business process reengineering efforts have occurred since the early Eighties, there is no consensus among industry analysts regarding their success or failure rates. The failure rates range from 40 percent to as high as 85 percent. These statistical variations point to the volatility inherent in BPR, and indicate that it is clearly a risky endeavor.

One of the major problems with BPR is that there are no templates or user manuals to follow. Organizations -- and the Navy in particular -- find security in the guidance of a user's manual that lays out each step one by one. Many organizations that would attempt to reengineer shy away from doing so for this very reason.

Since there is no user manual and the reengineering practice is fairly young it can be a risky venture. Many industry analysts have accused BPR of being too drastic, instead, they believe a more subtle approach is warranted. These analysts claim that BPR is valid for organizations that are on the brink of failure and are looking for a life vest. This is not the case, however: BPR can be used by any organization as a powerful tool to

ensure that the organization's processes are streamlined around ever changing market demands.

1. Principles for Success

Both successful and failed BPR efforts have many similarities and much can be learned from both. Foremost is the role that senior management can play. Success rates rise tremendously in those organizations in which senior managers are actively involved throughout the entire BPR process.

a. Senior Management Involvement

Before the process begins senior managers should establish the strategic planning and context in which the BPR effort will occur. To do so executives must define the business's goals both in the near and long term. This will help employees understand both their roles in the process and the need for BPR. Next, they should assist in defining the organization's business processes in much broader terms to include suppliers and customers as well as internal business units.

Senior management is also responsible for creating an atmosphere of creativity and openness in which all members of the organization feel free to participate. This can prove to be very challenging, since BPR efforts are frequently associated with downsizing and layoffs. Often BPR attempts are met with strong resistance from employees, and to soften the resistance management must clearly communicate why the BPR effort is needed and what can be expected from management.

b. Customer-centric

Any BPR undertaking must be customer-centric. To keep the customer at the center of the BPR initiatives the participants must continually ask themselves "How will the customer be better served ....?" realizing that the customer is concerned with better product quality, faster response times, flexible payment schedules, and competitive prices. With these goals in mind, the members in the process can look for ways to satisfy the customer's needs. Achieving any of these goals will bring direct value to the

customer and thereby increase customer satisfaction. Most BPR initiatives are attempted because the company has lost sight of its original goal of serving customers. Usually profits have fallen and senior management is compelled to turn to reengineering in order to stay afloat.

c. Start Small

Although the goal of BPR is wide-sweeping change, the expertise needed to make this change must often be acquired. A good way for organizations to gain BPR experience is to start with a small portion or one process of the company. Starting small does not mean that only one department is reengineered. Instead, it means focusing on reengineering just one business function that may span over department lines. The intent here is to acquire some experience and understanding regarding the dynamics of BPR that will contribute to higher success rates on future BPR attempts.

d. Information Technology

Finally, and probably the most important element of a successful BPR effort is the critical role that information technology (IT) can play. Information technology can be a tremendous leveraging point if utilized properly. Many analysts are still uncertain regarding the potentials of IT, but all will agree that it is critical to any BPR initiative. [Ref. 4, p. 56]

Information technology broadens the realm of possibilities available to BPR. BPR allows IT to be used in new ways, and tailors technology to fit the specific needs of the company. The goal is not automation, but to use IT to create new and more effective processes. Although reengineering efforts may be accomplished without the use of IT, IT is really the point of leverage that offers virtually unlimited design opportunities. IT gives those involved the tools to expand on the number of possible business solutions, and new ways of looking at their processes. IT tools such as work-flow applications, mobile communications, process design techniques, CASE tools, and modeling tools empower the BPR efforts. [Ref. 7, p. 6]

As previously stated, IT should be viewed as more than just a means of automation. IT should be viewed in a recursive relationship with BPR that will not necessarily end with the latest round of reengineering tries. It is believed that the most competitive organization will be those that are able to implement changes to their processes as the demands from the customers and market change. [Ref. 14, p. 12] To do so these organizations must view IT and BPR in a recursive fashion, with each being the key to thinking about the other, as Figure 2 illustrates.

In some cases an organization's IT architecture can be a road block to successful BPR. This may be the case when the existing architecture cannot be modified in the required time frame, or when the IS department is inflexible to any new changes. Likewise, the existing infrastructure may not be able to support the proposed process changes, or may not be constrained by financial limitations. Managing these issues may prove to be as challenging as the reengineering effort itself.

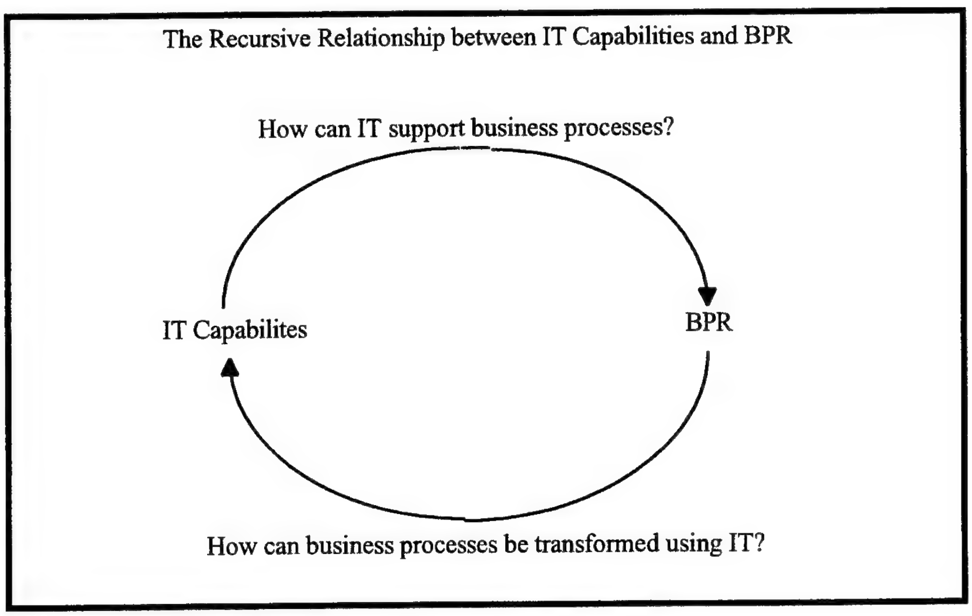

Figure 2, From [Ref. 14, p. 13].

The temptation is to design IT to support individual departments business functions rather than corporate-wide processes. Process improvements should not be constrained by the present capabilities of the IS department or by the limitations of the current IT architecture. Those involved in process redesign should not have any limitations but should be given an open plate to imagine as they will. Reengineering projects that have this type of open-ended atmosphere will be the ones that propose the best solutions.

The new challenge facing BPR and IT is the demand by companies to develop flexible, team-oriented, work environments. [Ref. 14, p. 12] Rather than maximize performance of individual business functions, companies want to maximize interdependent business processes across the entire organization. These processes should offer a new approach to doing business and will require a new computing model to support these new demands. Fortunately, there is indeed a new computing model surfacing replacing the old system, and enabling this new work environment to be a reality.

III. DOWNSIZING INFORMATION SYSTEMS

There has been an ongoing debate among industry analysts regarding the future of the mainframe. While many analysts believe it has outlived its useful life and is being replaced by client-server systems, there remains a smaller percentage who believe the mainframe will retain a niche in corporate computing. The attraction of client-server systems results from its ability to provide capabilities users demanded of mainframe systems but were not provided -- namely, graphical user interfaces, desktop application development, and real-time access to data. Consequently, many IS personnel have been swept up by the client-server trend and have attempted to migrate their systems without understanding the many implications involved in downsizing applications and platforms.

As the results of these "migration initiatives" were reported through the media, a remarkably low percentage of migrants reported any kind of success. The initial assertions made by industry analysts of lower operating costs, increased user productive, cheaper application development costs, and faster access to corporate data proved illusive. In actuality, migrating to client-servers was met with huge up-front costs, a lack of knowledgeable and experienced systems personnel, interoperability problems, and user frustrations. Consequently, industry analysts have posed a more thoughtful approach and have warned of the hidden pitfalls of migrating to client-server systems.

The push to migrate off the mainframe has been coupled with a "shoot the mainframe" mentality. Much of this mentality comes from pioneers such as Michael Hammer, who has coined the saying "Don't Automate, Obliterate," referring to business process reengineering and corporate computing. This mentality has subsided somewhat as it has not offered corporate mainframe users with any real alternatives to which they can entrust their critical applications. The mainframe has been the processing lifeblood for years, represents years of investment, and cannot be simply turned off for the latest industry trend.

However, with the rise of personal computing, end users have grown increasingly more frustrated with the short-comings of the mainframe and they, like industry analysts, have demanded a change to the computing model. For both sets of people, the days of dumb terminals are over as the advantages of deploying distributed systems can no longer be ignored. [Ref. 15, p. 28] The promise of these advantages has sparked the latest revolution in today's computing environment: downsizing.

Downsizing may be defined as the migration of traditional mainframe applications to smaller, less expensive platforms. The challenge in downsizing is enabling these smaller and often distributed systems to function as a whole, in order to process the work normally managed by a central mainframe computer. [Ref. 16, p. 26] Although this trend seems to be prevailing, the mainframe is far from dead. Most believe that its new role will be as a part of the new client-server architecture.

A. MAINFRAMES

Mainframe computers refer to those computers exemplified by the family of IBM computers introduced in the early Sixties. These platforms were the dominant systems until the early Eighties when desktop computers were introduced and pressures were put on vendors to provide a new direction. Mainframes tended to be large and expensive, with operating systems that were very complex. [Ref. 16, p. 23]

Current, mainframes are characterized by possessing hundreds of MIPS of processing power, gigabytes of storage, I/O controllers, memory buffers, intelligent queuing capabilities, and high I/O bandwidth used to support large data sets and many users. [Ref. 17, p. 50] Mainframes can be differentiated from mini-computers based upon the number of terminals they support, backup and recovery methods and security practices. In general, mainframes support 200 or more terminals while minis support less than 200. [Ref. 18, p. 57]

1. History

The mainframe of today has its origins back to the Mark 1, built by IBM in 1943. The Mark 1 read its instructions from punch tape and was used to perform scientific calculations in research labs during the second world war. Its components were not electronic as they are today but were mechanical and electromechanical, employing vacuum tube and transistors as active elements. Electronic digital computers came later and were grouped into generations or "lines" based upon their underlying technology. Table 3 shows the evolution of IBM mainframes lines through 1991.

YEAR	COMPUTER	COMMENTS
1946 - 53	IBM 701, 702	First-generation computers, Used electrostatic storage
1953 - 59	IBM 650, 704, 705, 709	Late first-generation computers, used magnetic drum storage for main memory.
1959 - 64	IBM 7080, 7090, 1400	Second-generation, used transistors
1964 - 69	IBM 360	Third-generation computers, initiated a common architecture, notion of a line.
1969 - 80	IBM 370	Fourth-generation, introduced virtual storage and a sophisticated OS.
1981	IBM 370/XA	Continuation of 360 architecture
1991	IBM 390	Continuation of 360 arch.

Table 3: Evolution of IBM Mainframes, After [Ref. 19, p. 19]

During the Sixties IBM came out with its third line of computer systems, the IBM 360. The 360 line proved to be a huge success since they were the first "lines" to contain what would become IBM's standard architecture. By the late Sixties IBM had become the dominant vendor of mainframe computers credited mostly to their development and use of a standard architecture that ensured backward compatibility to earlier IBM systems. As users outgrew their systems, IBM offered more processing power in the 370 and later 370XA lines. These lines of computers had similar architectures and operating systems, and used extended versions of the same instruction set as their 360 predecessor. [Ref. 19, p. 18]

Since the late Eighties total mainframe sales revenues have declined steadily each year. This statistic would seem to indicate that the mainframe is headed for extinction, but this has not been the case for two reasons. First, mainframes have become much cheaper to build. Component parts cost considerably less to manufacture and assemble so total revenues have dropped in part because prices have fallen steeply. In 1980, mainframe MIPS cost about $400,000. The price dropped to $117,000 by 1989, and to $34,000 in 1994. Second, stiff competition from distributed systems, and other forms of computing, such as alternative mainframes, have forced manufactures to pass costs reductions to customers. Table 4 illustrates this point. [Ref. 20, p. 62]

Year	Mainframe MIPS Worldwide	Price Per MIPS
1987	236,234	$145,641
1988	346,857	$128,511
1989	464,437	$116,828
1990	604,406	$95.505
1993	911,975	$80,750
1994	1,009,097	$34,000

Table 4: The Price of Mainframe MIPS , After [Ref. 20, p. 62]

2. Purpose Served

The mainframe has served organizations well and will continue to be the central platform in organizations that require large data processing and transaction processing capabilities. For example, the airline industry's reservation system requires large amounts of memory, large databases, and processing power that is still only offered by a mainframe. This type of application is too large to be deployed on smaller systems, as the technology of smaller systems has not matured far enough yet. No doubt the airline industry, and others like it, will continue to employ the mainframes into the distant future. From a management point of view the mainframe has provided information systems personnel with centralized management capabilities. Mainframe management tools enabled IS personnel to read hardware diagnostics, monitor system status, fix problems on-line, and take preventive measures against failures without leaving the central computer room. This structure has mirrored the centralized corporation with its top-down management style.

Maniframe computing can no longer be justified from a dollars-to-MIPS ratio since this advantage is held by today's PCs. However mainframe computing continues to be popular based upon the operating economies of scale that still exist. These large computers still provide capabilities not otherwise available from smaller machines for example high I/O bandwidth and systems security.

3. Architecture

The mainframe architecture is conceptually rather simple and quite easy to model. Figure 3 gives a generic representation of a standard IBM mainframe computer. At the heart of the mainframe is the host processor which is responsible for controlling all hardware and software operations. In doing so, the host processor directs and manages the performance of both the front and back-end processors. These two processors are responsible for controlling data flow in and out of the host processor, so that the host processor can concentrate on system control and application processing.

The back-end processor serves the function of retrieving data from the various storage devices. Although this processor is shown as a separate unit in Figure 3, it is really a function of the mainframe's software. The software will determine which of the commands require the services of the storage devices and then will "off-load" that procedure to the appropriate storage device.

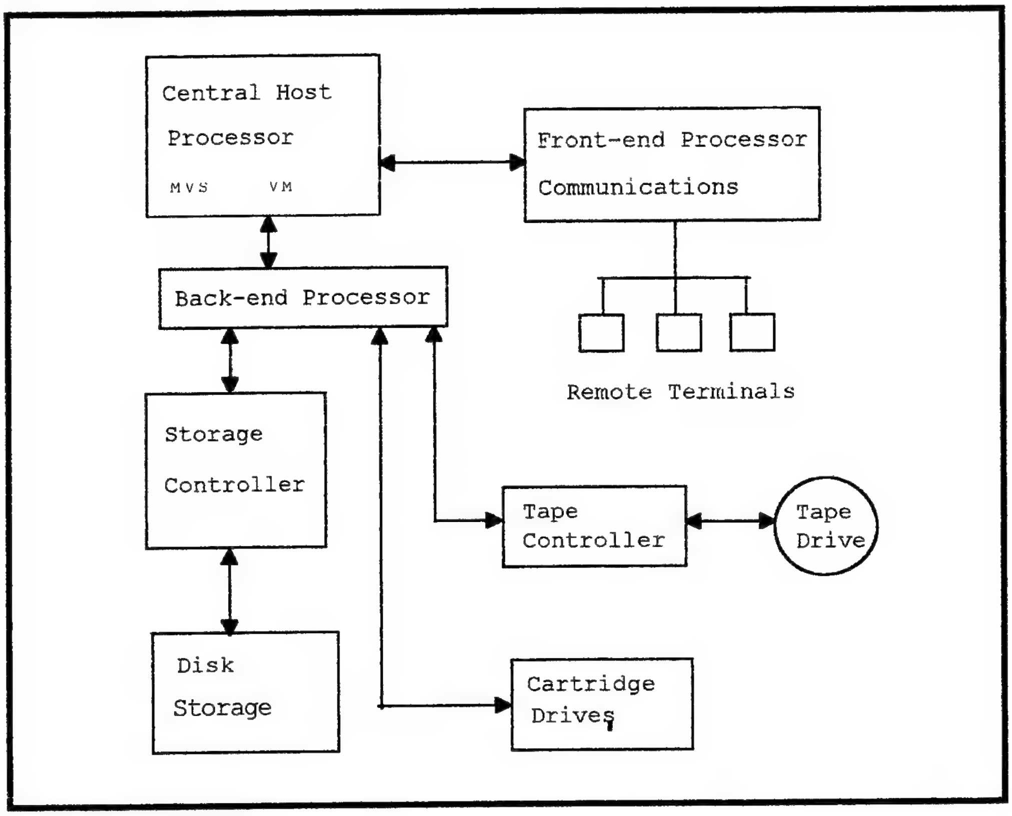

Figure 3: IBM Mainframe Architecture, From [Ref. 21]

Mainframe storage systems are very sophisticated incorporating a variety of storage devices, and storing data based upon the frequency in which the system accesses this data. Information or data that is accessed frequently is stored on smaller and faster access devices such as magnetic disks. Data that is used infrequently is stored on magnetic tape or cartridges. Most mainframe computers will employ all of these storage devices, and they comprise much of the physical space that a mainframe occupies.

Finally, the front-end processor assists the host processor by providing communication services to and from remote terminals. These remote terminals may be located in the room next door or in a separate geographical location. Use of both the front and back-end processors increases system efficiency. [Ref. 18, p. 57]

a. New Alternative Mainframe

The traditional mainframe has undergone substantial transitions within the last decade. Falling hardware costs, the rise of PCs, and increased use of network computing have caused mainframe vendors to modify the role of mainframe computers. Due in large part to the amount of negative press surrounding mainframes, vendors have shied away from calling them mainframes, and have instead referred to them as "parallel enterprise servers" or "large database servers." These new "alternative mainframes" have shown themselves to be just as powerful as traditional mainframes. Some of these new alternative machines employ up to seven processors side by side which allows them to be more nimble and efficient, while retaining all the advantages of traditional mainframes. [Ref. 22, p. 6]

This alternative mainframe has enabled IBM to retain mainframe computing in networked environments by fulfilling a role as central file and large database servers. These smaller machines often equal traditional mainframes in reliability and data storage capacity, and are scaleable to meet rising user's and network demands. Their low cost is now less than $10,000 per MIPS, while the annual operating cost is less than 90 percent of the traditional mainframes. [Ref. 23, p. 7]

4. Advantages of Centralization

Although there has been an attempt to "shoot" the mainframe, mainframe manufactures have provided new life via the alternative mainframe, and through efforts to maximize performance of older legacy systems. These efforts have led to a mainframe renaissance among those who realize that distributed technology is not yet mature enough to faithfully handle large scale critical applications. Joe Vincent, writing in "Computer World," states that the traditional mainframe out-performs other systems on most

accounts. He believes the mainframe is the only platform capable of performing heavy-duty, large volume processing chores. [Ref. 24, p .57]

For years the mainframe has been a dependable platform upon which users have entrusted their businesses processing. These users are a part of a mainframe culture that has deep roots back to the origins of computing. Unfortunately, in today's confusing computing environment many who migrate to downsized systems will miss the numerous advantages inherent in mainframe computing. The following list highlights some of these advantages mainframes hold over other systems.

- Economies of scale
- Architectural control
- Centralized asset management
- Centralized IS spending
- Centralized application development
- Security
- Back up and recovery
- Better management of IS personnel
- High data integrity
- Experienced operators and management specialists
- System robustness [Ref. 22, p. 8]

Some of these advantages could arguably be the short-comings of centralization. For example, application development has tremendous payoffs when it is accomplished by end users using today's application development tools. Likewise, better management of IS personnel may not be having them all centralized, but out in the business units working directly for the business managers developing applications that will help them. Increasingly users are seeing the wisdom in migrating to distributed systems as businesses become more decentralized and desire the flexibility to follow market trends. It remains to be seen, as many industry analysts have claimed, that the mainframe is a dying platform. [Ref. 25, p. 1]

B. ARCHITECTURAL SHIFT

When IS shops were centralized, management of computer systems was relatively easy to deal with. There were only a few platforms, and even fewer operating systems. Application programmers mastered one or, at most, two languages. IS staffs consisted of specialists who concentrated their efforts in one particular area. Information technology was manageable because it was one computer, one operating system, a few programmers, scheduled runs, and centralized management. IS Management was easy because the systems were homogenous and centralized.

This is not the case anymore. Nowadays there are a variety of computing platforms and operating systems. Applications are developed in many different languages, and without the help or permission of central IS personnel. IS personnel are no longer centralized but are dispersed throughout the organization and report directly to the business managers in the units in which they work. Every facet of managing Information Systems has undergone substantial change. The good old days of centralized mainframe management are gone. Although some organizations continue to operate mainframes, they usually do so within the context of a network. Clearly the domination of the mainframe has yielded to the new computing model of networked systems. Mainframes are no longer the dominant platform, or the platform of choice.

1. The Manageability Problem

Information technology became unmanageable during the late Eighties when the architecture shifted from centralized to decentralized systems. With the decentralization came heterogeneity consisting of multiple computers, multiple operating systems, multiple applications, multiple programming languages, and multiple databases. Staying abreast of these market shifts became virtually impossible. These shifts occurred as users voted with their pocketbooks against the mainframe in favor of personal computers. Users wanted more than system "up-time" and they found that networked PCs gave them this. Users demanded access to corporate data, developed applications, and performed operations that were once reserved for IS personnel.

The market has sought to provide the users with all that they crave for: more from their hardware, software, and operating systems in the form of faster processing speeds, larger memory, larger storage, faster printers, open systems, vendor independence, plug-and-play technology, application portability, and simpler more intuitive interfaces -- all at lower cost. These increased demands coupled with falling hardware costs have created tremendous shifts within the computer industry and made managing IS much more difficult.

While the trend has been to hide system complexity from the end user and provide easy interfaces in the form of graphical and object-oriented interfaces, the associated underlying complexity has made managing these systems much more difficult for IS personnel. IS personnel must confront issues of interoperability and application portability on a day-to-day basis -- issues that were unheard of in the good old days of mainframe management. IS managers are faced with the difficulty of enforcing system standards on heterogeneous and distributed systems throughout the entire organization instead of centralized systems located within just the IS shop. Management of an IS shop was at one time "do-able"; now it is orders of magnitude more difficult. [Ref. 26, p. 1]

2. The New Synthesis

The architectural shift is primarily the result of heterogeneity. Out of this turbulent period of transitions has surfaced what the Gartner Group calls the "new synthesis." They believe that the computer industry is on the threshold of a new era of manageable heterogeneous networked systems. Accordingly they believe that:

> A number of desperate, small advances have occurred during the past five years which successfully address some challenges inherent in integrating diverse systems. These advances are the beginning of a "New Synthesis," a collection of tools and techniques whose goal is single image network computing. The New Synthesis is more than a melting pot of modern software and networking technologies. It represents a paradigm shift form processor-centric methods of organizing computing toward a software-centric world view which organizes computing resources around software frameworks. This New Synthesis acknowledges a multivendor world for software and hardware... Processing in the new

era spans different hardware architecture's, different operating systems and many different types of middleware products. [Ref. 26, p. 1]

The new synthesis is software centric and is no longer one CPU to many users or applications, but many CPUs to many applications. The idea of a central processor has vanished. Systems implementers do not seek homogeneity but harmony among the many network modules. The new IS architecture groups the PC, applications, and the user at the center. In this new architecture the user is in charge and IS personnel play an assisting role. [Ref. 26, p. 1]

3. Target Architecture

This new synthesis or target architecture that most organizations will strive to achieve is a flexible three-tier system with a graphical user interface, and the security and reliability of a mainframe at the center. This three-tier hierarchy will consist of a desktop PC connected, via a network operating system, to a middle layer of applications and database servers, which will be further connected to a mainframe at the top. This model has the advantage of using each platform for what it does best. PCs offer low cost desktop processing providing users the autonomy and access to corporate data that the mainframe-workstations model was unable to do. The servers facilitate the sharing of resources between users and provide effective communication and control. The mainframe will provide maximum I/O performance, security, and manageability for the centralized data. This model may fold into two or possibly one as technological advancements continue. It is out of this new synthesis and the promises that it holds that the trend to downsize has occurred . [Ref. 17, p. 50]

C. DOWNSIZING MAINFRAMES

In its simplest form downsizing can be thought of as the downward migration of business applications from mainframes to smaller platforms. [Ref. 27, p. 7] The downsizing process breaks up large mainframe-type applications into separate modules that run will on one or more network servers where they are more suited for business and organizational needs. Successful downsizing requires thoughtful planning and must be

executed within the context of the overall corporate strategy. Business Process Reengineering is really an attempt to eliminate queues amongst tasks that have many hand-offs, and downsizing enables an organization to realize these improvements. To ensure the largest amount of success, any downsizing endeavor should be performed in conjunction with business process reengineering.

The press is filled with stories regarding the benefits of dismantling the mainframe and deploying distributed systems. The majority of these stories claim client-server systems as a big success providing cost savings and increased worker productivity over mainframe systems. A smaller percentage of the articles warn of the pitfalls of migrating to client servers, and question the data supporting the professed advantages. In spite of these warnings, downsizing to client-servers appears to be the prevailing industry trend. A 1994 survey by Forrester Research found that of America's top 100 largest companies, 65 percent were already using client-server systems and another 15 percent had pilot programs underway. By the end of the decade, client-server computing will likely be the norm for most companies. [Ref. 20, p. 62]

One of the few analysts who is slow to jump on the client-server bandwagon is Paul Strassman, who states that:

> the problem in measuring the effects of decentralization is finding enough corporations that have reported on their decentralization moves. Today's views have banished the centralized MIS organization along with the mainframe. Instead the distributed setup is supposed to offer the most effective solution. That may happen someday, but last year's numbers don't support the view that productivity and decentralization are synonymous. [Ref. 28, p. 83]

The downsizing process should be approached cautiously, and the entire corporate climate should be assessed to evaluate if downsizing is the right course of action. If downsizing is the proper choice, the next step is to develop a migration strategy that fits into the overall corporate strategy. A survey of 400 major corporations conducted by the Gartner Group in 1993 found the three top reasons for downsizing were: 1) the potential for increased functionality afforded to the user, 2) the enabling of

business process reengineering, and 3) application reengineering. [Ref. 23, p. 10] Downsizing mainframe applications will pose many challenges for the IS manager. The following discussion frames many of these challenges.

1. Downsizing Challenges

It has been commonly subscribed to that mainframes are the only safe and practical computing platform for mission critical applications. Recent advancements in computer platforms and software capabilities have shown that this remains true for a shrinking number of computing situations. As client-server systems achieve greater credibility, more and more organizations will be willing to entrust their critical applications to them. However, this transition may be slower than client-server vendors would like for merely economic reasons. Organizations cannot afford to dismantle the centralized mainframe environment that has required such a huge investment. Organizations giving thought to downsizing should not plan on migrating to client-servers overnight, but should plan on a more orderly "creep" to this new computing model. [Ref. 26, p. 31]

Knowing the challenges associated with downsizing the mainframe and migrating applications to smaller platforms will enable Navy IS managers to increase the likelihood that the migration strategy will be a success. Orchestrating a successful downsizing strategy is much more difficult than deciding on a target architecture or adhering to downsizing mandates such as "implement all applications using open systems" or "move everything to PC LAN's today." What is required is thoughtful planning of the implications that the *process* will entail.

Evaluating these implications and thoroughly analyzing the many critical issues will increase the probability of the project's success. The four areas the downsizing plan should focus on are management issues, software applications, hardware considerations, and cost factors. The following sections contain questions that help to frame the downsizing process. These lists of questions are not intended to be exhaustive, but rather assist in assessing the downsizing process.

a. Management Issues.

Management has the largest responsibility within the downsizing project since management controls each phase. Paramount is the responsibility to ensure that any downsizing of information systems is accomplished within the overall business strategy. Downsizing information systems for the wrong reasons can be a costly mistake. Unfortunately, there are too many examples where downsizing initiatives were performed because it was assumed that cost savings would result, or because all neighboring corporations were doing it. Management must also ensure that IS personnel are involved in the planning process to avoid the risk of having highly fragmented LANs established, with business units creating their own solutions independent of corporate strategies. [Ref. 29, p. 5] Questions that management should address are:

- Does the plan to downsize fit in with the overall corporate strategy?
- What will be solved by downsizing?
- Is the corporate climate conducive to downsizing?
- How healthy is the IS shop?
- When is the appropriate time to downsize?
- Will the IS department need outside help?
- Will the corporation need change management consultants?
- Will the corporation standardize applications throughout all business units or allow users to use whatever suits their needs?
- Will the new applications meet the corporation's business needs for the next two to three years?
- What can be done to anticipate the next wave of demands?
- Is the company risk averse? [Ref. 23, p. 34]

b. Software Issues.

Management will want to stay in touch with the intentions of the IS department to ensure that IS decisions complement the organization's strategy. Any new information systems that are the result of a downsizing initiative should allow the

organization to be as flexible as possible. For this reason open systems will probably be the best and most flexible option for the organization. The following questions should be addressed by the IS department and reviewed by management:

- Who needs what information (data) and when?
- Which of our current applications will need slight or heavy modifications?
- Can commercial off-the-shelf (COTS) software satisfy the corporate computing needs?
- Which applications will be migrated and in what order?
- Are there applications that cannot be migrated? Will there be a need for additional programmers?
- What will the development or conversion timetable be?
- Will the downsized applications require more or less maintenance?
- Will there be a standardized GUI?
- Will business units be allowed to develop their own applications?
- What types and amounts of middleware will be needed?
- What will be the new data model and will individual business units need to share data?
- What will be the new database?
- Does the target platform support the programming language, DBMS, or middleware.

c. *Hardware Issues.*

As already mentioned, hardware issues will be addressed by the IS department in conjunction with software concerns. IS personnel should shoot to have the hardware be as flexible as possible to allow for adaptations as the market changes. The following questions should be considered:

- Will the target architecture be an open system?
- Can the current workstations be used?

- Will the mainframe continue in service? If so, for how long, and in what capacity? If not, will the migration be a turn-key evolution or will the implementation be done in parallel?
- Are IS personnel familiar with the new platforms?
- Will training of IS personnel be needed?
- What state of the art hardware will meet corporate needs for the next three to five years?

d. Cost Considerations.

Probably no single issue associated with downsizing has received more attention than the cost savings. It has been assumed that any migration to smaller systems equates to cost savings. This is just not true. Costs associated with deploying client-server technology are deceptive and industry feedback is contradictory. Initial beliefs were that migrating to client-server systems would produce immediate cost savings, but more recent results have shown this to be untrue. In light of these more recent results, organizations must be ready to absorb the up-front costs associated with deploying a new system. For IS department that have years of managing centralized operations, distributed systems will pose new challenges. Network management is a relatively new discipline and there is a shortage of experienced personnel. The following questions should be addressed in an attempt to get a ballpark figure for the cost of deploying client-server systems.

- Will reducing the workload on the mainframe produce cost savings?
- Will the promise of worker productivity associated with development of desktop applications be allusive?
- What can be the expected time-table for return on investment?
- What will be the costs of software development or reuse? How much will training of IS personnel and users cost?

Initially industry analysts were claiming that mainframes were not only outdated but expensive to operate and maintain compared to distributed systems. There still exists a false belief that migrating applications to smaller systems will save the organization money, both in the short and long run. According to the Gartner Group the costs of switching an application from a mainframe to a new platform was under estimated because planners overlooked some critical considerations applications conversion and maintenance, data integrity, network issues, and operations and administration. The following list breaks down these four areas into more detailed tasks.

- Application program conversion and maintenance
  - retraining development staff on the new operating system and new middleware
  - transferring the application code to the new environment
  - compiling, modifying, and recompiling application programs
  - re-testing (often the biggest part of this project)
  - re-documenting
  - retraining end users
- Data
  - transferring data to the new platform, translating data types and formats as necessary
  - cleaning up the data so that it will meet the integrity constraints of the new DBMS
  - writing extract and update programs to keep files on the old and new platforms in synch
  - running regular reconciliation jobs (uploads and downloads)
- Network
  - regenerating an existing network, substituting new hosts, or
  - installing new networks, new terminals of PC's and new controllers

- Operations and administration
    - hiring new staff or retraining the current staff for duplicate environments
    - selecting, purchasing and installing new system and network management tools [Ref. 26, p,.33]

Downsizing has not proven to be a straightforward endeavor. Many organizations have become much more hesitant to downsize now that they realize that the costs are more than they had originally anticipated. These costs have soared largely because labor-related costs have sky-rocketed. For example, since 1987 PC administration costs have more than quadrupled and end-user operations costs have doubled . Although technology-related costs have fallen about 30 percent annually, the drop has not been enough to offset these hefty labor-related costs. Furthermore, weak migration planning has also helped drive up downsizing costs. [Ref. 30, p. 6,] In the mainframe environment the major costs were related to physical assets: large mainframes, peripherals, operating systems, and a variety of application software. In moving to a distributed system, management and control functions have shifted from the centralized center to user departments where the major cost is labor.

2. Identifying Applications to be Downsized

The essential element in any successful downsizing project is to carefully understand the workload being performed by the mainframe. The proper downsizing approach and target architecture to be selected may be known only after the mainframe applications are fully understood. Each mainframe application must be considered separately to determine whether and how it will be moved to the new platform. This will allow different applications and portions of applications to be migrated as the planning team sees fit. To get the most out of the downsizing efforts each application must be considered in light of the corporate strategy; looking five to ten years down the road and anticipating organizational shifts. [Ref. 26, p. 31] Each system application should be evaluated for size, performance, complexity, and condition. Likewise, consideration should be given to the value versus the cost of migrating an application. Those

applications that would produce a cost savings if downsized, are given a higher priority in the migration planning. Once each application is assessed, the fate of the mainframe will be more clearly known. These four considerations are highlighted below:

- size of applications
  - scope
  - count of on-line transactions and modules
  - count of batch processes and modules
  - size of largest batch procedure, largest program, and large sub-routine
  - count of database tables and views
  - count of files and record types
- performance considerations
  - transaction throughput
  - transaction response time
  - scheduled and required up-time
  - batch window and volumes
  - data volatility
- complexity issues
  - essential complexity
  - accidental complexity (multiple processors, languages, databases)
  - interfaces to other applications
  - interfaces to other systems or special equipment
- applications condition
  - age
  - code modularity, structure, and consistency
  - quality of the system documentation [Ref. 31, p. 48]

a. *Poor Candidates for Client -Server Systems*

Unfortunately, not all applications are suited for client-server deployment. The two basic motivations for computing are automation and empowerment. Automation involves replacing people with technology while empowerment augments people with technology. Mainframe systems are very good at automation, while client-server systems have not yet reached the level of maturity where they can be entrusted with automated production-line processes. Client-server systems hold advantages in information display and "any time" availability, where traditional mainframes have been weak. Applications that are poor candidates for client-server platforms are systems characterized by at least one of the following four categories: very large and complex systems, systems with large centralized I/O processing, the need for centralized control, and tight mainframe integration. [Ref. 31, p. 37]

(1) System Size and Complexity. Large transaction processing systems are for the most part poor candidates for a GUI/database client-server approach. Usually these systems require rote repetition from the users, in which case the user interface is often simple enough that it does not need to be graphical. In addition, very complex systems are not good candidates for client-server technology unless the system can be broken down into very discreet and logical components. [Ref. 31, p. 37] Client-server systems are by themselves a difficult undertaking; adding to this a complex application only compounds the difficulty. Likewise, applications in which thousands of users share a common database should probably be left on the mainframe. Possible bandwidth and traffic problems may occur over the communication lines. System management for these types of applications is better tackled by the traditional centralized data center.

(2) Large Centralized I/O Processing. Large databases, on the order of 20 gigabytes, that cannot be partitioned should remain on the mainframe as should systems with large batch requirements. Client-server systems have not yet matured to the level where they can be entrusted with these types of systems. [Ref. 31, p. 37]

(3) Centralized Control. Systems that require any type of centrally managed control such as security or other important services should be left on the mainframe. Client-server systems by their design are not conducive to central management and while it is true that some level of centralization can be achieved, the cost of doing so is often prohibitive. Those client-server systems where centralized management is advantageous are smaller systems with 50 workstations or less.

(4) Tight Mainframe Integration. If an application is tightly integrated with other mainframe applications, it is going to be difficult to migrate. It may be possible to off-load some processing, but if the shared data has to be communicated or replicated, there may be no benefit in switching to a client-server system. [Ref. 31, p. 37]

b. Good Candidates for Client-Server Systems

As a general rule systems that empower their user such as executive information systems, decision support systems, and systems that allow for ad hoc queries are all best designed using a client-server approach. Other situations ripe for client-server systems are financial, mathematical, and statistical analysis, CAD, medical engineering and software development work. [Ref. 31, p. 39] Systems that do not fall into these classifications may still be candidates for client-server systems, however these classifications have a proven history of performance as client-server applications.

3. Downsizing Strategies

There are many approaches to downsizing, and no doubt numerous companies are undergoing downsizing efforts because they believe their mainframe applications are no longer useful and need to be replaced. In order for downsizing efforts to be successful organizations must take into consideration the overall business strategy, and address the basic issues regarding the business goals. It is not recommended that all mainframe applications be scrapped simultaneously and the organization attempt a one-time effort to replace all existing applications and systems. Some reengineering champions such as Michael Hammer actually advocate this type of approach. More important is that the organization understand the nature of the current mainframe workload and then make

migrate plans accordingly. Paul Kavanagh in his book, Downsizing for Client-Server Applications, recommends that for each application the following series of questions should be addressed:

- Is the basic business process necessary?
- Is the application working well?
- Is additional functionality needed?
- Is additional ease of use needed?
- Is the underlying technology working well?
- Is the technology expensive or obsolete? [Ref. 31, p. 41]

Once these issues have been addressed, it is then possible to determine the outcome of each application and what the organization will do with it. For any given application the organization can choose between one of six courses of action:

a. Remove the System.

This approach may involve abandoning or outsourcing the business function or even doing it manually. Software applications are usually difficult to maintain and the associated cost with maintaining them is very high. It is therefore worthwhile to consider whether the current application should exist at all. Some applications are around because they were initiated by someone who still holds power in the organization and is opposed to scrapping them. Others are maintained because no one has ever taken the time to question their existence. [Ref. 31, p. 21]

b. Replace with Packaged Software.

This approach will be possible if the business process being supported is not unique to the company but performed by others in the industry. Usually larger companies have the resources to build their own applications while smaller companies run largely on packaged software. Two recent trends have occurred that have made commercial off-the-shelf software a more viable solution for more organizations. First, the packages have become more powerful, easily customized, and accessible from other applications. Second, the larger organizations have realized that their business functions

of accounting, inventory management, and human resources are not so different that they require custom code. These new applications offer increased functionalities to the point where organizations are willing to redesign their processes around these state-of-the-art packages. [Ref. 31, p. 23]

c. Rewrite the Software.

This approach requires constructing a replacement system using current application development tools. This approach is often used when the old business process has changed substantially and the current application has become obsolete, and a commercial off-the-shelf replacement cannot be found. Then the most viable option is to rewrite a new application. [Ref. 31, p. 25]

d. Rehost the Existing Software.

This approach will entail modifying the current application and moving it to a new platform. Fortunately, there are a number of products available that can run the same application code on another platform. Applications that are afforded this luxury are therefore good candidates for downsizing. [Ref. 31, p. 27]

e. Refurbish the Existing Software.

This approach will involve leaving the application on the current platform, while improving its appearance, use, or maintainability. Organizations will want to refurbish their existing systems when it appears these systems will continue to meet the needs of the company for the next couple of years, or if the business logic is contained within the current system and the organization achieves some sort of competitive advantage from the application.

Refurbishing the code may involve modifying the user interface, business rules, or database systems. Refurbishing the user interface can be done by using a tool that improves the appearance without changing the original application code. Refurbishing the database will involve cleaning up the data and making it accessible to

other applications. Making it accessible may involve migrating to a relational database or using timed replication to copy the data to various repositories.

Refurbishing the business rules will almost always be required for applications that have been around for many years. These applications have experienced a large amount of decay and in most cases no longer support the business process they were intended. Quite often the company has grown up around these applications and matured in spite of them. Eventually these applications become outdated and the underlying business rules must be reengineered before any modifications can be made. [Ref. 31, p. 28]

f. Surround.

This approach will involve developing a new environment while retaining the application and the data on the old system. This makes sense since it is more cost effective to develop new applications on new platforms while leaving old applications on old platforms. This technique attempts to hide the old application by surrounding it with a newer more intuitive application. [Ref. 31, p. 30] After planning the fate of each application, the IS department and the organization's management will be able initiate the downsizing of applications that fits into the overall corporate strategy.

4. Critical Success Factors

The critical success factors of any downsizing project revolve around the commitment of top management to the downsizing project, and the IS shop's ability to prepare the organization for the new technology. Top management must be fully supportive throughout the entire process, and not only during the initial phases. If the project falls upon hard times the commitment of top management will indicate to all the level of importance attached to the project and its completion. Management must be open to failure and not shy about new projects and the pitfalls they may possess. Management must also show enthusiasm of the new client-server architecture and the possibilities it holds for the organization. Finally, communication from senior

management is essential as it will calm employee fears about the application and the implications it might have over their jobs.

The second factor that contributes to a successful downsizing project is the IS shop's ability to prepare the organization for the new application and system. This may include additional training, placing IS personnel in the business units where they are more accessible to end users, and continual communication between IS leadership and the end user community. Communication from the IS shop contributes significantly to any project's success. IS personnel hold the unique responsibility of ensuring that the end user community is adequately trained so that end users do not get overly frustrated from the onset regarding the application. [Ref. 27, p. 61]

Other critical success factors include a phased migration plan that will allow for the implementation plan to be accomplished in incremental steps rather than a turn-key transition, obtaining a second opinion from a consultant firm or another IS professional outside of the organization, and assessing the state of the organization's culture and its resistance to change. If the company has a poor rate of success regarding change initiatives, then management might want to rethink the downsizing strategy.

a. Business and Technological Assessment

The key ingredient to any downsizing project is to ensure that the system being developed is solving a business need. Applying the right technology to the wrong problem results from management's inability to accurately assess the business situation. As stated earlier many organizations are downsizing to save on computing costs. This trend should be avoided since recent findings show costs actually increase over the long term in client-server architecture.

Properly assessing the current business environment should include identifying and prioritizing the business problems, evaluating the organization's installed technology, identifying the customers and the competition, and identifying the market share held by the company. Most of these assessments are relatively straightforward and can be accomplished rather easily and objectively. However, with regards to assessing

the current technological infrastructure, the feedback provided by the current maintainers will often be exaggerated regarding system complexity and value. For this reason it is best not to give too much weight to the opinions of the current maintainers. [Ref. 31, p. 47] These people will in all likelihood be unable to give objective evaluations of the systems value to the organizations. They run the risk of believing that the system or application that they maintain is of greater importance than it actually is.

b. Risk Assessment

All downsizing efforts are subject to risk, and like all change initiatives downsizing must be approached cautiously. Unfortunately, many believe that downsizing projects involve migrating to smaller machines, and therefore the risks are less because smaller machines are understood by a larger percentage of the user population. This erroneously assumes that there is much less to go wrong than there would be with a larger and more complex mainframe. However, in downsizing any application the sources of risk are often hard to identify and difficult to manage. Sources of risk include the end user, the technology being implemented, and the organization. [Ref. 32, p. 76] The following list of questions will help identify the sources of risks that may arise among these three key areas.

- End User
    - What's the amount of impact on the users?
    - How much change will they experience?
    - Are user requirements clearly known?
    - What's the user's relationship to the IS department?
    - Is the IS department viewed favorably or unfavorably?
    - Do the users understand the new technology?
    - Are the users opposed to change?

- Technology Implementation
    - Is the technology too complex for the organization and current level of user knowledge?
    - Is the new technology the wrong technology for the business problem being solved?
    - Will the users understand the application?
    - Will the application fit the user's business needs?
    - Is the user married to the old technology?
- Organizational Climate
    - Is the organization's leadership stable?
    - Are there frequent management changes?
    - Are there frequent organizational and directional changes?
    - Is the IS department viewed as weak?
    - Is the organization undergoing a BPR process or other type of quality initiative simultaneously?
    - What is the political climate in the organization?
    - Is this downsizing project the "pet" of one company officer?

    [Ref. 31, p. 32]

Likewise, consideration should be given to the risks associated with not downsizing. Electing not to downsize may prove detrimental to the organization's competitive position. Not deploying an application may allow a competitor to gain an advantage in customer response time or in delivering a product to the market.

Whatever the risks, it is imperative that the risk assessment process be a large part of the work accomplished before the downsizing effort. The ability of an organization to manage these risk elements will to a large extent determine the success of the downsizing effort. These risk elements represent the critical success and failure factors associated with any application or system migration.

c. *Successful Downsizing Projects*

A good downsizing project will look a lot like any well-managed project. As mentioned numerous times throughout this chapter, success revolves around the ability of senior management to stay supportive of the program, the ability of the IS department to implement the transition, and the "goodness of fit" of the application or suite of applications that will be deployed on the new architecture. The following lists, while not exhaustive, contain critical success factors that if managed properly will increase the probability that the project is implemented safely.

- senior management enthusiasm for the client-server architecture and the possibilities it holds for the organization
- realization that implementing client-server systems will cost a lot of money up-front
- business processes that are customer-centric
- positive organizational climate that is not resistant to change
- willingness of IS staff to receive training and acquire new skills
- migration plan that fits into the overall corporate strategy
- get a second opinion from someone outside the organization
- selection of the right application to meet the business solutions
- IS department does its homework regarding the technical capabilities of the new system
- phased migration
- continual communication throughout the process [Ref. 27, p. 61]

Naturally any of these critical success factors could be turned over and be viewed as failure factors. Downsizing is a risky undertaking as many industry analysts are now pointing out. The myth of cost savings associated with downsizing has just about been shattered, as the results of the latest migrations show. It is imperative that the managers take all the success and failure factors into consideration before committing to downsizing their system. A careful evaluation of the size, performance, complexity, and condition of each system and application can help eliminate hasty decisions and ensure

that the company is pursuing the best possible strategy for its successful operation and growth.

IV. CLIENT SERVER SYSTEMS

A. EVOLUTION OF CLIENT SERVER TECHNOLOGY

Client-server technology was the inevitable outcome of several trends impinging upon the work place during the late Eighties and early Nineties. As users grew increasingly more frustrated with the limitations of the mainframe they turned to other forms of computing. The computing model that most end users turned to was the local area network that permitted the sharing of resources among end users. However, these local area networks were often isolated from one another and were incapable of allowing end users to access corporate data located on the mainframe. Eventually end users became frustrated with the limitations of these local area networks.

Meanwhile additional trends among businesses were to reengineer their business processes and downsize mainframe applications. Both of these trends revealed the value that information technology, and namely client-server systems, could provide as a link to both the past in mainframes and the future in distributed systems. These new client-server systems promised to employ each computing platform for what it did best: the mainframe would continue to provide large centralized processing capabilities, and the clients would serve as flexible platforms allowing end users the freedom to create applications as business needs dictated.

Eventually client-server systems were seen as a means to leverage the enormous potential of the isolated LANs located throughout most organizations. Client-server systems held the potential to allow these isolated LANs to communicate across different protocols and transmission mediums giving the user a single image view of these connected networks. The promise of one network, where all users would be able to share network resources and data, became one of the most sought after features of client-server systems. Additionally client-server systems were the only viable option to the traditional mainframe and its many shortcomings.

B. WHY DEVELOP CLIENT-SERVER SYSTEMS

It was discussed previously that the new computing model of heterogeneous systems created a manageability problem for IS departments. In fact, the viewpoint held by most in the computer industry is that managing client-server systems is a much more challenging task than managing traditional mainframes. The management tools available to client-server network administrators are not as mature or sophisticated as those afforded to the traditional mainframe managers. Furthermore, the problems associated with dispersed IS personnel, end user application development, network security, data integrity, and hardware and software maintenance make client-server systems appear less attractive to IS departments.

So why would any organization want to migrate to, or deploy client-server systems? The answer lies in the tremendous benefits afforded to the end users. Client-server systems provide end users with capabilities they were unable to obtain from traditional mainframe computing, namely increased access to corporate data, desktop processing, application development, and the sharing of resources. Furthermore, deploying client-server systems allows an organization to be employ the various computer platforms, located throughout the organization, in a capacity that most suits the platform's strength. The following list highlights some of the many reasons why organizations deploy client-server systems:

- it makes downsizing possible
- provides easy and transparent access to corporate data
- more efficient use of corporate computing resources
- scaleable architecture
- application development at the desktop
- reduced application development backlog
- establishment of an "open" system architecture
- empowered end users [Ref. 33, p. 12]

C. WHAT IS CLIENT-SERVER

There is quite a bit of confusion surrounding what constitutes client-server computing. A lot of this confusion stems from client-server vendors claiming to provide solutions to every distributed computing ailment. Addressing the myriad of client-server issues can be unsettling and often raises additional questions such as: Which products are client-server and which ones aren't? Must all clients have a graphical user interface in a client-server environment? Can an application be client-server if it isn't built with client-server based products? Can a desktop PC be both client and server? This chapter will discuss these and other similar issues.

Basically a client-server system is one developed so that parts of it can run on separate computers. [Ref. 31, p. 1] The key to understanding this definition is realizing that client-server is a logical concept. That is, client-server refers to an application and not a hardware configuration. Usually, for an application to qualify as a client-server application, it must have been developed to run on different systems. This is not to imply that all applications must run on separate machines, but they must have the *capability* to do so. [Ref. 31, p. 95]

As a system model, client-server enables the interaction between software processes that are executing simultaneously on different machines. Cooperation between the client and the server exists through messages sent back and forth between the two. As the name implies, servers provide services to their clients, usually in the form of specific processing that only they can do. By off-loading processing chores to the servers, clients are free to process other tasks until the results are received back from the server. In a true client-server environment, both the client and the server processes can be located on the same or different machines. [Ref. 5, p. 3]

1. Application Architecture

Most business applications can be broken down into three separate layers: the user interface layer, the application logic layer, and the data management layer. [Ref. 31, p. 99] The critical element in deploying any client-server application is deciding how the

second layer, or the application logic, is to be distributed over the different computing platforms that make up the network. This decision will allow the application to take advantage of the strengths of the various platforms that comprise the client-server system. Figure 4 illustrates the three layers of most business applications.

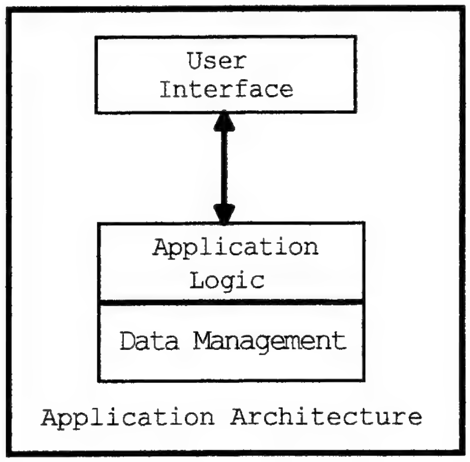

Figure 4, After [Ref. 35, p. 288]

a. User Interface Layer

The user interface layer is also known as the presentation layer. This layer accepts and presents information to the user on the screen. While a user interface doesn't necessarily have to be graphical, the graphical user interface (GUIs) is the most commonly used type. The GUI is responsible for providing the user with an efficient way to understand the functioning of the application, and will hopefully remove the fear associated with learning new applications. David Vaskevitch in his book Client-Server Strategies "credits the GUI with being the primary reason why PC use has become so widespread throughout the world." [Ref. 35, p. 288]

b. Application Logic Layer

The layer below the user interface layer is the application logic layer. This layer enforces the business rules of an organization which are the operations and procedures around which the business functions. [Ref. 35, p. 291] The Gartner Group

has defined five models of client-server computing based upon how the application logic is spread out over the network. Figure 5 illustrates these five different client-server models: distributed presentation, remote presentation, distributed logic, remote data management, and distributed database.

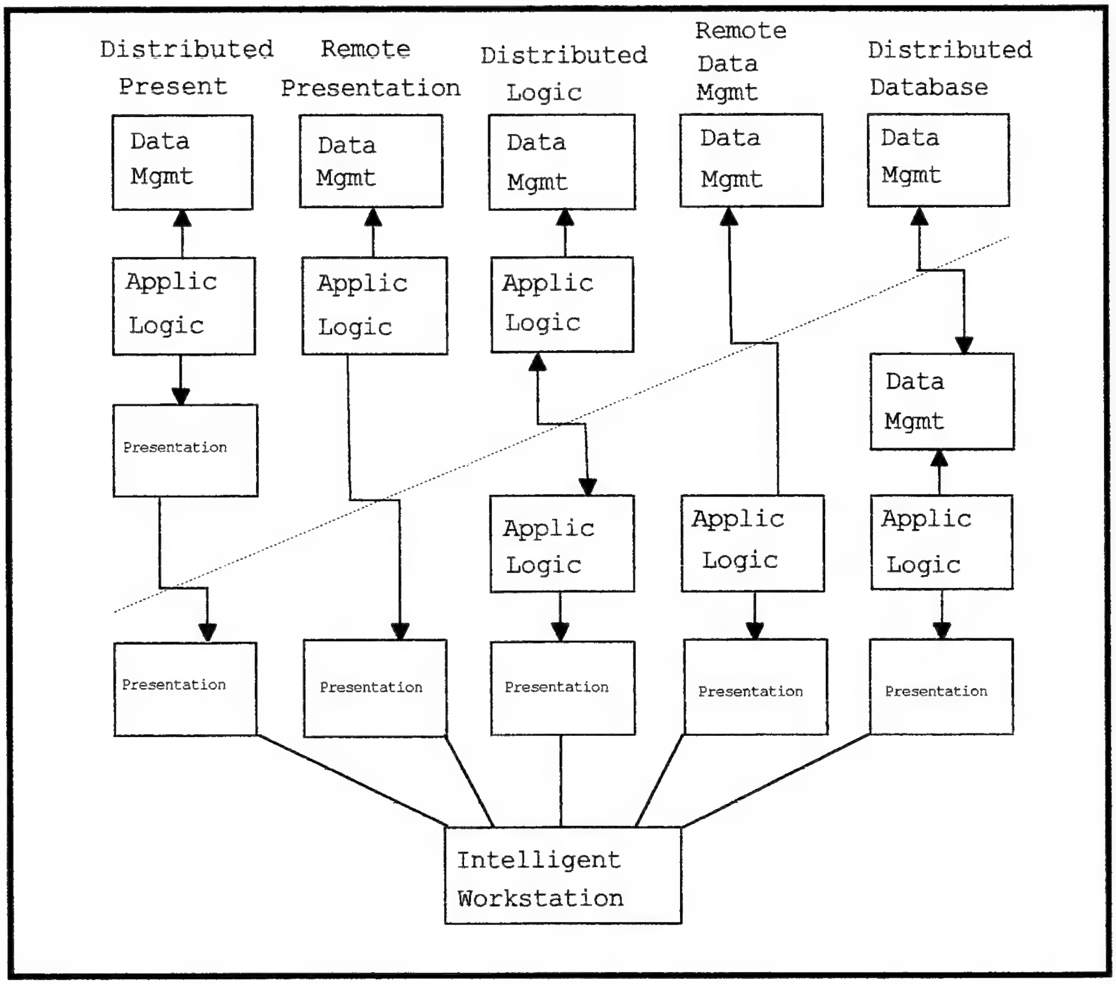

Figure 5: The Five Types of Client/Server Computing, After [Ref. 34, p. 11]

c. *Data Management Layer*

The third layer is the data management layer. This layer is primarily responsible for maintaining secure and consistent data through the use of database management systems. The database management systems are responsible for data storage

and retrieval, maintenance of database records, and data integrity. [Ref. 35, p. 290] Table 5 lists the functions of the three layers of the application architecture.

Layer	Responsibility	Functions	Tools
Interface Layer	Understandable, efficient interface	Presentation, navigation, manipulation, analysis	Graphical tools and languages
Application Logic Layer	Policy: rules and heuristics	Decision making, policy enforcement, and resource coordination	C, COBOL, rule processors, BASIC
Data Management Layer	Consistent, secure data	Consistency, security, integrity, and safety	Databases, database languages

Table 5: Functions of the Layers in the Application Architecture, After [Ref. 35, p. 287]

2. Open Systems

The uncertainty surrounding the term "open systems" stems from the fact that there are various degrees of openness. In a system that is truly open, hardware and software components from any vendor can be removed and replaced by components from any other vendor. One of the best examples of an open system is the 386/486 PC. These machines can be assembled using components from a wide variety of vendors. Ironically, the openness of today's 386/486 originated from a system that was built with components that were nearly monopolies of Intel processors and Microsoft operating systems. However, through various industry wide organizations, standards were established that helped produce more open systems. [Ref. 31, p. 10]

The advantages of open systems are that they afford the users interoperability, portability, and scalability. If systems are open they will allow the users the ability to scale their systems as business needs dictate. It is only logical that systems built with

openness in mind will allow the organization to remain flexible enough to adjust with market shifts.

At the opposite end of the spectrum are closed or proprietary systems. Closed systems are those whose standards are not known to the general public, but are controlled by only one vendor. Closed systems have the disadvantage of being limited in functionality to those services that one vendor can provide.

One of the biggest challenges facing IS personnel is integrating the many different platforms, applications and operating systems that comprise the typical computing environment. Heterogeneous computing is the standard, and the challenge is to ensure as much cross platform compatibility as possible. A system is considered open to the degree that it allows heterogeneous systems to communicate with each other.

3. Scalability

Scalability implies that a system can be "right-sized" to larger or smaller systems as necessary. Scaleable systems should support interoperability standards, so that the data kept on one system can be accessed from other systems. Scalability usually goes hand in hand with openness as these systems will allow its users to upgrade the systems as business needs change. [Ref. 31, p. 13]

D. CLIENT-SERVER BUILDING BLOCKS

The building blocks of client-server systems are: the graphical user interface (GUIs), network operating systems (NOS), middleware, and database management systems (DBMS).

1. GUIs

As mentioned previously, one of the goals of the GUI was to make the computer more intuitive to first-time users. Prior to GUIs, users were required to learn cryptic text-based commands in order to manipulate an application. Not only were the commands cryptic, but few applications, even among the same vendors, had similar commands. The learning curve, required for each new application, was often the

determining factor that caused new users to give up on computers, and kept experienced users from learning new applications. [Ref. 35, p. 72]

With the widespread distribution of GUIs, computers became more intuitive to first time users. As GUIs became more standardized among different vendors and applications users who had mastered one application had to a large extent mastered them all. GUIs were credited with providing the user with the same look-and-feel across vendor and application boundaries. David Vaskevitch, in his book Client-Server Strategies, credits the GUI being the primary reason why computers have been so well accepted by end users. He states that:

> The common user interface of a GUI defines a standard way of commanding the computer to do things. The user of pull-down menus, coupled with a help system, enables the user to explore the application, literally discovering commands often without having to read any documentation. Furthermore, because all applications use the same broad structure, after learning how to use that first application, the user has, in many ways, learned to use them all. [Ref. 35, p. 73]

2. Network Operating Systems (NOS)

To a large extent network operating systems evolved to respond to the inherent limitations of Microsoft's DOS. MS-DOS was designed for the stand-alone PC and was not intended to meet the needs of networked systems. As users began to network their PCs many did so in spite of the limitations of MS-DOS. As these little isolated LANs took shape it was apparent that a NOS with multitasking capabilities was needed to coordinate the sharing of resources and communications between users. [Ref. 31, p. 120]

Network operating systems have been greatly increased the use of network computing by allowing communications between stand alone computers. Network operating systems coordinate the exchange of computing and data resources located throughout the organization, and allow for a more efficient use of network resources. Over the last few years the lines between desktop operating systems and NOS have blurred. Operating systems such as the Apple Macintosh, UNIX, OS/2, and more recently Windows NT can serve as both the desktop OS and the NOS.

Network operating systems can be thought of in terms of the base and extended services they provide. The basic services are file and printer sharing, and to a smaller extent system security. The extended services include global directory services, fault-tolerant file storage, and a variety of management capabilities. [Ref. 31, p. 121] It is these extended network services that are so critical to the efficient functioning of a client-server system, and it is from these services that a user is provided with a single image of the network.

a. Communications

Network communication between clients and servers occur by either remote procedure calls (RPCs) or through the use of message passing. In both communication schemes it is the responsibility of the network operating system to hide the details that make communications between clients and servers rather complex. This discussion will focus on the logic concepts of network communications and leave the more detailed topics of protocols, synchronization, and address resolutions to a separate cover.

In message passing a server needs to be able to determine which client sent the message, since a server can receive a message from any of a number of different clients. To send a message a client process executes a generic *send(message, to destination)* system call to a server. A server in turn, queues the *send* in a port, where it can store multiple *messages* from the many clients it serves. Once the server has processed the *send* message, it returns the results to the client via a *send(Reply1, Client1)*. [Ref. 5, p. 163] This message passing scheme is illustrated in Figure 6.

The second form of network communications is through remote procedure calls (RPC). RPCs are similar to calling a local procedure, but in this case, the RPC executes the procedure on another platform. When a client process executes a RPC, the local process in suspended, the calling parameters are sent to the remote procedure's location, and the procedure is executed there. When the remote procedure completes the process, the results are sent back across the network, and the calling process resumes

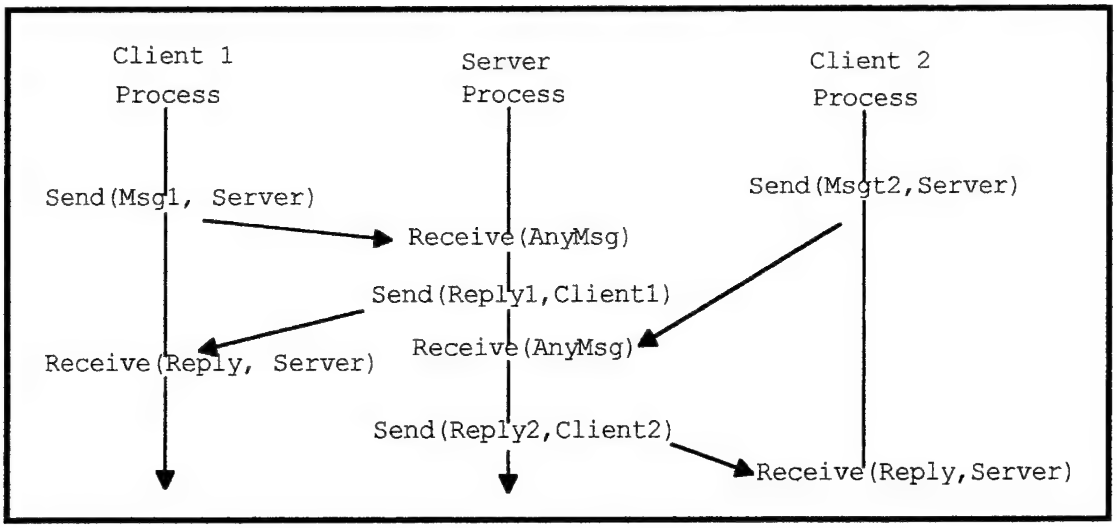

Figure 6: General Message Passing, After [Ref. 5, p. 164]

processing as if it were returning from a local procedure call. A RPC is viewed by the client as if it were executing the procedure locally. In this process the program is suspended while the client passes to the server the parameters of the RPC and then waits until the result is passed back. [Ref. 5, p. 175] This concept is illustrated in Figure 7.

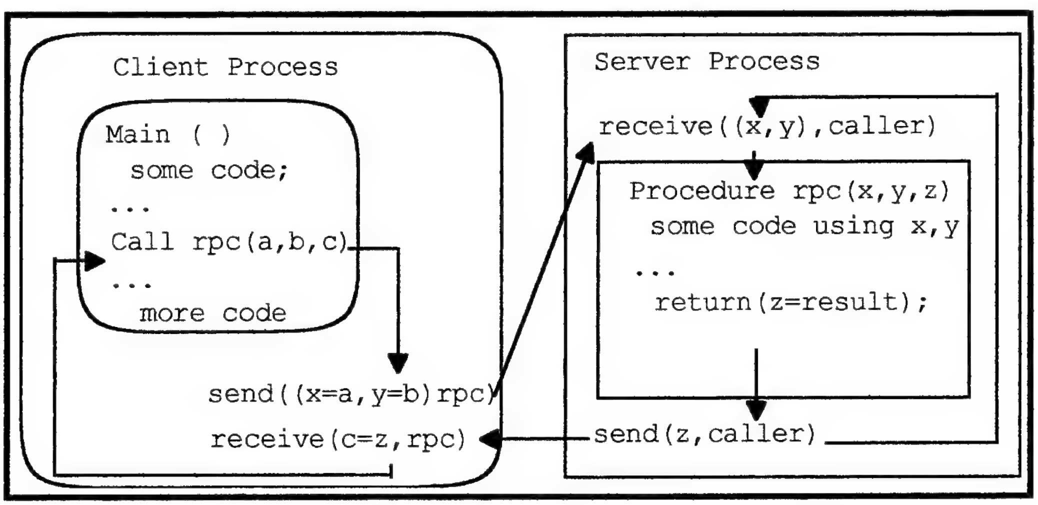

Figure 7: Remote Procedure Call, From [Ref. 5, p. 175]

3. Middleware

In its broadest definition middleware can be defined to include all the distributed software required to support the interaction between clients and servers. [Ref. 36, p. 18] Middleware gives the user the impression that the entire network functions as one system. Through the use of middleware different clients are able to communicate with different servers seamlessly, and the user views his workstation as being the entire network. To a large extent it is middleware that enables a client-server system to be considered open.

4. DBMS

The job of managing the organizations' data is handled by the database management system (DBMS). Today's DBMSs use a relational data model and a data manipulation tool called Structured Query Language (SQL) to manage the data contained in the organization's databases. DBMSs have continued to improve and have matured into rather eloquent Relational DBMSs.

One of the basic goals of a DBMS is to allow an organization to improve the use and control of its data. To accomplish this, a DBMS will provide data integrity, data security, ease of use, and data accessibility. There are trade-offs from choosing among these various objectives such that concentrating on one will often be at the expense of one of the other capabilities.

Two problems continue to impede the future development of DBMSs. The first is the inability to design a DBMS that allows several users to access and update the same data simultaneously. When this situation occurs one user is locked out while the other accesses the data. The second problem has to do with distributed DBMSs in which the synchronized update of distributed data located throughout the network becomes an issue. Global locking and two-phase commit are mechanisms that attempt to address this issue but to date they are incapable of providing effective distributed databases employed by OLTP systems.

E. CLIENT-SERVER APPLICATION DEVELOPMENT TOOLS

There are a large number of development tools available to application programmers in today's computing environment. Many of these tools are focused on converting mainframe applications into client-server applications, or providing a GUI overlay for an existing mainframe application. A subset of these development tools are those specifically designed for the development of client-server applications. These products usually contain the ability to rapidly develop prototype applications that allow users a more interactive role in the development process. The following discussion will focus on these tools designed for the client-server environment.

There is a subset of Fourth Generation Languages that use object technology to rapidly develop client-server applications. Some believe these tools will become the Fifth Generation Languages and provide the means to develop applications using visual building blocks. Products such as Borland's Delphi allow developers to build applications using pre-packaged components which can be visually combined into complete applications.

The real power of these tools stems from their library of components which allow developers to assemble applications with connections to databases, video, imaging, and messaging. These tools enable the programmer to rapidly develop a working model from which the end user can begin to provide feedback to the developer on application functionality. [Ref. 16, p. 13] This reiterative process is referred to as Rapid Application Development (RAD) and promises to greatly improve the software development process that for years had been stymied by rigid methodologies.

1. Fourth Generation Languages (4GLs)

Software prototyping really gained momentum with the introduction of 4GLs. These tools are really more than programming languages, but may be viewed as programming environments. As such, they offer the programmer a complete package of programming tools. [Ref. 16, p. 12] Most 4GLs contain the following tools:

- DBMS
- Data Dictionary
- Interactive query facilities
- Report generator
- Screen formatter
- Word processor and text editor
- Graphics
- Library of macros
- Programming interface
- Reusable code
- Software development library
- Backup and recovery
- Security and privacy safeguards
- Links to other DBMSs [Ref. 6, p. 267]

4GLs offer an excellent environment in which software applications can be prototyped on the end user's desktop. Prototyping gives the end user an opportunity to evaluate the application together with the developer, thereby by-passing the rigid procedures of earlier development methodologies. Thus, rapid prototyping gives the end user an application quickly, which in turn allows iterative feedback to be given to the developer so that greater amounts of functionality can be included into the program. [Ref. 6, p. 269]

2. CASE Tools

CASE tools were first used by mainframe programmers developing large and complex applications. CASE products were aimed at automating code generation during the structured methodology practices that existed during the mainframe era. CASE may be defined as any automated tool that assists in the creation, maintenance, and/or management of software systems. [Ref. 6, p. 273]

Although CASE tools were originally designed for large applications running on mainframes, CASE vendors have made a shift toward client-server applications. These CASE products claim to offer the benefits of the original mainframe CASE packages but, as of yet have failed to be widely accepted by the application development community.

F. CRITICAL ISSUES

The initial claims that migrating to client-server systems would save organizations money was based upon the falling prices of personal computers (PCs). PC prices were falling on the average of about 30 percent per year. Therefore, a logical conclusion was to assume that any migration away from the mainframe toward PCs would produce tremendous cost savings. The initial migrants off mainframes were chasing after these allusive savings. Unfortunately, most of their ambitions stemmed from hype generated by industry analysts and vendors alike.

As the results of these early migration initiatives filtered back from these pioneers, the claims of lower costs seemed to be inaccurate. In response to this new data industry analysts took a more conservative stance and claimed that migrating to client-server systems contained many hidden costs that could not have been originally predicted. It has since been accepted that client-server systems are often more expensive than their predecessors, the mainframes over the long-term.

1. Client-Server's Hidden Costs

The major hidden costs revolve around network management and labor-related issues. The network management issues basically concern the management of data for integrity, availability, recoverability, and security. Accomplishing these goals requires multiple levels of storage devices, network access measures, and safeguards against disasters. Finally there are the human costs of managing the network to ensure the system receives the proper level of service it requires. [Ref. 37, p. 6]

From an IS department point of view client-server computing equates to increased management problems over the traditional mainframe. However, the benefits from these new systems are that they empower the end user, which in turn will provide the

foundation to allow the company to be more competitive. The bottom-line on client-server computing is that it helps companies make money in areas of the company outside of the IS department. According to Diane Tunick of the Gartner Group, the benefits from client-server systems are that they:

> empower employees throughout the enterprise by giving them immediate and transparent access to information. As a result, companies can strengthen their competitive stance, enhance customer service, shorten time to market and streamline their staffs. Moreover, client-server computing is a logical fit for the profound organizational changes most companies are undergoing, ... [Ref. 37, p. 6]

a. Support and Training Costs

It is surprising that industry analysts overlooked the support and training costs associated with client-server systems and later referred to them as hidden costs. Making the transition from mainframes to distributed systems is a major shift in the computing model that would naturally be accompanied by large training costs. End users would be confronted with unfamiliar interfaces and methods of accessing data in ways they were not accustomed to. Likewise, application development would be entirely different as would system maintenance and management. Therefore, it is more startling that the industry analysts missed these obvious costs and placed so much value on falling hardware figures.

One of the leading hidden costs is the overhead resulting from non IS personnel performing computer related maintenance for other office workers. When an end user seeks technical assistance from another office worker, who is known as the "resident computer expert," this individual can spend much of his or her time trouble-shooting another user's system and thus be less productive at his designated job. These local "experts" become the resident trouble desk, and distort the actual costs of running the IS trouble desk. As end users enjoy more success at fulfilling their own trouble-shooting needs, the less likely they are to turn to the established trouble desk. As

these technically savvy end users become more relied upon by other office workers they will have a big impact on the hidden costs associated with IS support. [Ref. 38, p. 3]

Finally, due to the diversity of applications and the configurations of LANs and desktops, support costs can be furthered strained through the required skill level of the technicians needed to support these different platforms. [Ref. 39, p. 65] With a centralized mainframe support shop technicians were accustomed to supporting end users in a limited number of ways, either by trouble-shooting non-responsive keyboards or wholesale keyboard swap-outs. This limited range of skills would not begin to meet the maintenance needs of today's diverse and heterogeneous office environment.

b. Management Costs

To support the client-server environment new management challenges surface such as local area/wide area network administration, trouble desk management, and training programs. Due to the diversity of components in most client-server systems, managing these systems becomes a staggering task. Additionally, network management tools have not achieved the level of sophistication of those possessed by the traditional mainframe community which increases the management costs associated with client-server systems.

In order to minimize management costs an approach to network management should be applied to reflect the level of complexity of the deployed client-server system. One way to accomplish this is through the use of standards that will ensure that all the components of a client-server network will be interoperable. This will also help minimize system complexity.

Another way to help limit system complexity is to restrict the systems to a few products that have proven to work together. Some industry analysts believe that a smart procurement practice is to purchase systems that are two to three years behind cutting edge technology. This will provide time for vendors to work the bugs out and give the market time to indicate which product line is "best." Along this same vein is the idea of selecting an application suite that will integrate well with the network operating

system and application development tools. All of these practices will help reduce client-server management costs by providing easier methods to support the network infrastructure. [Ref. 40, p. 17]

G. THE NEW COMPUTING MODEL

The creation of client-server computing was inevitable. It resulted from the changing business environment, the diversity of products on the market, and the demand by end users for access to corporate data and interconnectivity. Additionally, client-server systems were attractive because they promised to provide those benefits that the mainframe had been unable to offer. As the various components of client-server systems such as GUIs, NOSs and DBMS matured, the benefits of migrating to client-server systems became even more pronounced. Organizations were willing to migrate off the mainframe in spite of the fact that doing so meant increased costs.

The IS model for most large organizations today is a three-tier model with a mainframe or mini-computer at the top, followed by a second layer of various types of network servers, and a third layer of client desktop machines. The server layer is responsible for managing the network resources and coordinating the communications among the various layers of the model.

The beauty of this model is that it enables organizations with legacy systems to surround these systems with other platforms that can be used to access the mainframe's data. Organizations do not have to weigh the implications of scrapping the mainframe, but can elect to move in a somewhat orderly fashion toward a more distributed approach. Client-server systems not only promise to be the next computing model they remain to be the only viable option to traditional mainframes.

V. CONCLUSION

The goal of this thesis was to educate Navy IS managers regarding the major issues associated with downsizing information systems. The mandate to downsize information systems is firmly embedded in the minds of most senior Naval Officers. However, as this thesis has shown, not all information systems should be downsized. The responsibility rests on Navy IS managers to fill the gap between empty downsizing mandates and sound organizational decisions. To do so, the downsizing issues must be properly framed within the context of the larger industry-wide trends.

Chapter two was devoted to Business Process Reengineering (BPR). BPR is not a passing fad but will continue to play a strategic role in the reshaping of organizations and IS shops for quite some time. The goal of this chapter was to show that information technology is the enabling force behind any reengineering effort, as it allows for the creative thinking of process redesign and task consolidation. Without the use of IT most BPR efforts are severely limited.

Chapter three was the heart of this thesis. The computing model that has dominated the IS world for over thirty years has undergone a major transition from centralized mainframes to distributed client-server systems. This shift, referred to as downsizing, stems from the advantages afforded to end users empowered with a desktop computer as opposed to the traditional dumb terminal. In this new model the value gained from these desktop computers makes downsizing a very attractive and strategic endeavor for most organizations.

This chapter focused on the management issues associated with moving applications off the mainframe to smaller systems. Downsizing mainframe applications is not an easy task. There are no downsizing manuals, and experience with downsizing applications among IS personnel remains low. For this reason, it is important that IS shops gain some experience in downsizing before attempting to downsize mission-critical applications.

Consideration was also given to the risks of downsizing and the factors that ensure a successful downsizing process. Most downsizing projects are subject to risks that come from an organization's climate, the skill level of the IS shop, and the support of senior management. A key point made was the need for any downsizing initiative to be done in light of the organization's long-term strategy.

Chapter four discussed the components of a client-server system. The goal here was to eliminate some of the confusion surrounding what constitutes client-server systems and applications. This chapter also looked at some of the critical issues related to hidden costs and the management of client-server systems.

Hidden costs have plagued client-server computing from their introduction. It is now fairly well accepted that over the long-term client-server systems will not produce any cost savings over the traditional mainframe. Yet, as previously mentioned, organizations continue to deploy client-server systems because of the many benefits they promise to deliver.

Finally, there are many strains placed on IS personnel having to manage this new architecture. Some of these strains have originated from the fact that today end users play a more active role in the development and maintenance of information systems. Previously, IS personnel were responsible for every facets of the computing model. Now the end user has control over items such as application development and desktop hardware procurement. In the future IS shops will probably find that their role will consist of managing the IT infrastructure only.

In closing, the overall goal of this thesis was to highlight the relationship between Business Process Reengineering, downsizing, and the development of client-server systems. Since there is not much concrete guidance regarding downsizing mainframes this thesis has attempted to frame the issues and will hopefully provide Navy IS managers with a base from which sound management decisions can be made.

LIST OF REFERENCES

1. Tomasko, R.M., *Downsizing*, Prentice-Hall Inc., 1987.

2. Drucker, P., "The Coming of The New Organization," *Harvard Business Review*, , ,

3. Byrne, J.A., "The Horizontal Corporation," *Business Week*, 20 December 1992.

4. Hammer, M., and Champy, J., *Reengineering the Corporation: A Manifesto for Business Revolution*, HarperCollins, 1993.

5. Renaud, P.E., *Introduction to Client-Server Systems,* John Wiley & Sons Inc., 1993.

6. McNurlin, B.C., and Sprague, R.H., *Information Systems Management in Practice*, 3rd ed., Prentice Hall, 1993.

7. Gartner Group Strategic Analysis Report, *BPR: Radical Change, Dramatic Results*, by A. Cushman, and R. Terdiman, 25 May 1994.

8. Gartner Group Strategic Analysis Report, *Flattening Organizations - Four Tier to Two Tier*, by R. Smith, 6 January 1992.

9. Gartner Group Strategic Analysis Report, *IT and Business Strategy: A Research Summary*, by J. Jakobovits and R. Smith, 21 April 1993.

10. Hammer, M., "Reengineering Work: Don't Automate Obliterate," *Harvard Business Review*, July/August 1990.

11. Gartner Group Strategic Analysis Report, *Situation and Setting are Keys to Successful BPR*, by R. Smith, 6 December 1994.

12. Hammer, M., "Making the Quantum Leap," *Beyond Computing* March/April 1992.

13. Benjamin, R.I., and Levinson, E., "A Framework for Managing IT Enabled Changed," *Sloan Management Review*, Summer 1993.

14. Davenport, T. H., and Short, J.E., "The New Industrial Engineering," *Sloan Management Review*, Summer 1990.

15. Miller, J., "Client/Server Advantages," *Computer Weekly*, 21 April 1994.

16. Shark, J.D., *Novell's Guide to Client/Server Applications and Architecture's,* Novell Press, 1994.

17. Reinhardt, A., "Your Next Mainframe," *Byte*, May 1995.

18. Long, L., and Long, N., *Computers*, 3rd ed., Prentice-Hall Inc., 1993.

19. Prasad, N.S., *IBM Mainframe Architecture and Design*, McGraw-Hill Book Co., 1989.

20. Cook, W.J., "Serving Up a New Era in Computing," *US. News & World Reports*, 17 October 1994.

21. Gentry, B., Associate Director Computer Services, Naval Postgraduate School, Monterey California

22. Mullins, R., "Mainframe Computers - Innovations," *The Milwaukee Business Journal*, 1 October 1994.

23. Gartner Group Strategic Analysis Report, "*The Alternative Mainframe, Not Always an Alternative*," by Latham, L., and Tunick, D., 12 January 1994.

24. Vincent, J., "Downsizing Information Systems: Another Look," *Computer World*, 30 May 1994.

25. Gartner Group Strategic Analysis Report, "*Once More: What is the Best IS Organization Structure?*" by Raphaelian, G., and Terdiman, R., 31 January 1994.

26. Gartner Group Strategic Analysis Report, "*From Traditional Mainframes to the New Synthesis*," by Malik, W., Schulte, R., and Younder, E., 28 July 1993.

27. Natraj, N.D., "Downsizing Information Systems," presentation given at the Naval Postgraduate School, Monterey, California, 7 September 1993.

28. Strassman, P.A., "Centralization's Payback," *Computer World*, 6 June 1994.

29. Gartner Group Strategic Analysis Report, "*OIS in the 1990's: The Migration Dilemma*," by Anders, M., 6 August 1993.

30. Gartner Group, Inc., *Inside Gartner Group this Week: Downsizing Strategies for the 1990's: Part 4*, by Tunick, D., 7 July 1993.

31. Kavanagh, P., *Downsizing for Client/Server Applications*, Academic Press Inc., 1995.

32. Trimmer, D., *Downsizing Strategies for Success in the Modern Computer World*, Addison-Wesley Publishing Company, 1993.

33. Achultheis, R., and Bock, D., "Benefits and Barriers to Client/Server Computing," *Journal of Systems Management*, February 1994.

34. Gartner Group, Inc., *Inside Gartner Group this Week: Client/Server Computing Strategies for the 1990's: Part 1*, by Miller, C., 28 July 1993.

35. Vaskevitch, D., *Client/Server Strategies*, IDG Books Worldwide, Inc., 1993.

36. Orfali, R., Harkey, D., and Edwards, J., *Essential Client/Server Survival Guide*, Van Nostrand Reinhold, 1994.

37. Gartner Group Strategic Analysis Report, *Justifying The Cost of Client/Server Computing*, by Tunick, D., 7 April 1993.

38. Gartner Group Strategic Analysis Report, *Minimizing LAN Costs, Improving Support Yields Big Dividends,* by Cappuccio, D., and Younder, E., 23 September 1994.

39. Gurbaxani, A., and Kennedy, M., "Can the Mainframe Compete with Client/Server Systems on Cost?" *Telecommunications*, November 1994.

40. Toscano, S., "So How Much Does It Cost?" *PC Week,* 4 July 1994.

INITIAL DISTRIBUTION LIST

		Number of Copies
1.	Defense Technical Information Center Cameron Station Alexandria, Virginia 22304-6145	2
2.	Library, Code 52 Naval Postgraduate School Monterey, California 93943-5101	2
3.	James C. Emery Department of Systems Management Code SM/Ey Naval Postgraduate School Monterey California 93943-5002	2
4.	Myung.W. Suh Department of Systems Management Code SM/Su Naval Postgraduate School Monterey, California 93943-5002	1
5.	LT Todd L. Barnum JAST Program Office 1745 Jefferson Davis Hwy Suite 307 Arlington, Virginia 22202	2